A Case-Based Approach to Creative Design

ONR Annual Report: 1 Oct 92 - 30 Sep 93

Contract # N00014-92-J-1234

Principal Investigator: Janet L. Kolodner
Co-Investigator: Ashok Goel
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
(404) 894-3285
jlk@cc.gatech.edu

5 October 1993

Principal Investigator: Janet L. Kolodner
PI Institution: Georgia Institute of Technology
PI Phone Number: (404)894-3285
PI E-mail Address: jlk@cc.gatech.edu
Grant or Contract Title: A Case-Based Approach to Creative Design
Grant or Contract Number: N00014-92-J-1234
Reporting Period: 1 Oct 92 - 30 Sep 92

1 Productivity Measures

Refereed papers submitted but not yet published: 3

Refereed papers published: 4

Unrefereed reports and articles: 2

Books or parts thereof submitted but not yet published: 1

Books or parts thereof published: 1

Patents filed but not yet granted: 0

Patents granted: 0

Invited presentations: 8

Contributed presentations: 4

Honors received: 4

> Kolodner has been appointed steering committee chair for the Cognitive Science Conference to be held in Atlanta, GA in August 1994. She has also been acting as EduTech Institute interim director and has been selected to be a member of the steering committee for the proposed Engineering Research Center.
>
> Goel has been appointed a Vice-Chair of the third International AI in Design Conference to be held in Zurich, Switzerland in August 1994.

Prizes or awards received (Nobel, Japan, Turing, etc.): 0

Promotions obtained: 0

Graduate students supported ≥ 25% of full time: 2

Post-docs supported ≥ 25% of full time: 1

Minorities supported: 2

2 Detailed Summary of Technical Progress

We are studying and modeling creative design processes. Our goals are two-fold. One is to make intelligent, computer-based design assistants more creative (e.g., able to suggest unusual but useful solutions and to bring up important issues that might not have been considered otherwise). The other is to build computational models that help us understand human creativity. This will have implications for design education and suggest ways of enhancing the creativity of human designers.

2.1 Exploratory Study

To gain insights into the knowledge and reasoning involved in creative design. we performed an exploratory study of student mechanical engineers engaged in a seven-week undergraduate design project. In this study. we observed a great deal of the design process, including "official" as well as informal team meetings (e.g.. while choosing materials at a store). This has given us insights into the processes underlying many creative design activities. particularly the following. How designers generate alternative views of a problem through situation assessment and reformulation. How problem constraints and evaluation criteria gradually emerge or become refined as ideas are proposed and critiqued. How designers serendipitously recognize solutions to pending problems. often seeing new functions and purposes for common design pieces in the process.

2.2 Results of Study

Our study has found that creativity is not a process that gets turned on and off but arises out of a confluence of processes (such as problem elaboration and understanding, remembering. adaptation. evaluation and refinement of proposed solutions), each of which we all do everyday. and each of which interacts with the other processes in complex ways. Much of what we talk about as creativity arises from interesting strategic control of these processes and their integration. Thus. under our view. one doesn't talk about a creative person or even a creative product, but rather a creative *process*. Those of us with more interesting strategic control of our reasoning processes, including the ability to make connections between things. tend to reason in ways that produce more interesting results. (Our analysis of our observations is summarized in [Kolodner 1993a] and [Kolodner 1993b].)

Our model of the creative design process is shown in Figure 1. Creative designers often start with an incomplete, contradictory, and underconstrained description of what needs to be designed and transform it into something with more detail. more concrete specifications, and more clearly defined and consistent constraints. At the same time, creative designers generate several design alternatives, elaborating and adapting them. and often incorporating pieces of one into another.

It is the evaluation of these alternatives that is the core driving force behind these processes. The designer continually updates the design specification as well as a pool of design ideas under consideration. Each alternative generated is evaluated to identify its advantages and disadvantages and to check that it satisfies the constraints in the current design specification. A key part of evaluation is "trying out" the alternative (e.g., through experimentation or mental simulation). This generates a more detailed description of the alternative, including the consequences of its operation and how environmental factors affect it.

Evaluation raises questions of legality or desirableness of features of a design alternative and it detects contradictions and ambiguities in the specification. The resolution of these questions.

contradictions, and ambiguities serves to refine, augment, and reformulate the design specification. On the generative side, the critique generated during evaluation provides the basis for comparison of alternatives, often suggesting interesting adaptations or ways of merging them.

The three processes interact opportunistically. The generative phase, guided by critiques from the evaluation phase, watches for opportunities to merge or adapt design ideas to create new alternatives. The design specification is incrementally updated as ideas are tested and flaws or desirable features become apparent.

The continual elaboration and reformulation of the problem (i.e., the design specification) derives abstract connections between the current problem and similar problems in other domains, facilitating cross-contextual transfer of design ideas. Continual redescription of what the solution (i.e., the evolving design) looks like primes the designer to serendipitously recognize the solution if the designer comes across it. In other words, redescription creates a "lens" with which to assess new situations, enabling the designer to overcome functional fixation and see alternative functions and uses for common design pieces.

2.3 Case-Based Computational Model

These processes rely heavily on previous design experiences and knowledge of designed artifacts. An expert designer knows of many design experiences, accumulated from personally designing artifacts, being given case studies of designs in school, and observing artifacts designed by others. Through our observations and analyses we have found that reminding of these experiences is crucial to generating design alternatives, reformulating and elaborating the problem specification or proposed solutions, predicting the outcome of making certain design decisions, enabling visualization and simulation of proposed designs, and communicating abstract ideas in concrete terms.

The experiences that are most valuable are often highly contextualized pieces of knowledge about these artifacts, such as how a device behaves in some context of use, circumstances in which it can fail, and knowledge about situations that might come up not only in use, but in all phases of its life cycle. Given the nature of these experiences, we are using case-based representations and reasoning techniques [Kolodner 1993bk] to model the creative processes we have identified.

A particularly significant role that design cases play is in addressing the problem of *focus*: How does the designer know which details to pay attention to? Which aspects of an old design can suggest problem reformulations or can fill in missing details of the specification? During problem reformulation, which constraints should be relaxed or strengthened? Which evaluative questions and criteria should be raised to critique the proposed design options?

Design cases help address these issues by providing information about the consequences of past situations and what details were important in previous designs. Intentionally interpreting the current situation in terms of past experiences and reinterpreting previous solutions in the current context help to reveal and make explicit underlying assumptions. This can often lead to a useful problem reformulation or relaxation of constraints. (Details of how cases help address focus-related issues can be found in [Kolodner 1993b].)

We are also exploring the important role design experiences play in the theory development and conceptual change that occur in evolving a design specification. In our study, the student designers came to a better understanding of what the constraints of the problem were by performing many experiments with proposed design pieces and by recalling experiences they had had with devices for solving similar problems. These led to theories to account for the outcome of the experiments and previous designs. Sometimes an experiment or recalled case did not fit within an existing theory: explaining this anomalous data resulted in a conceptual change which led to a new way of viewing the problem to be solved. In general. theory development helps to refine vague. abstract problem constraints making them more concrete and operationalized.

Conceptual change involves a fundamental change in the underlying knowledge representations in terms of which the reasoner thinks about the domain. It involves the construction of new concepts and theories. and the modification and extrapolation of existing concepts and theories in novel situations [Ram 1993]. We are studying conceptual change not only in the context of specification evolution. but also in the context of story comprehension [Moorman 1993]. Consider. for example. reading a science fiction story. in which one must learn enough about an unusual world to accept it as the background for the story, and then must understand the story itself. In general. all types of reading - indeed. all types of comprehension - require us to learn about and modify our conceptions and beliefs to some extent. We have found that many of the same creative processes are involved in understanding unusual and novel situations as are involved in solving problems and designing in these situations.

2.4 Integrated Case-Based and Model-Based Computational Models

We are also studying integrated computational models that combine the use of design cases with the use of functional models for analyzing and modeling design processes. The functional models may be design-specific or design-independent. Design-specific models specify how the structure of a given designed artifact results in the achievement of its functions (e.g., how the functions of the components in an electrical circuit get composed into the functions of the circuit as a whole). while design-independent models represent how a causal process results in a specific behavior (e.g.. how the process of heat flow results in a change in temperature). In our earlier work we showed that functional models can provide answers to several issues in case-based design, e.g., they provide a vocabulary for indexing designs cases in memory (model-based indexing). an array of repair plans for adapting a past case to meet new design specifications (model-based adaptation), and a method for evaluating a candidate design (model-based evaluation).

In our current work we are building on this theme to model the processes of creative design. A key characteristic of creative design is the discovery of new design constraints in the process of evaluating a candidate design. The discovered constraints lead to a reformulation of the design problem because they introduce new design variables into the design problem space. Prabhakar and Goel [1992] have shown how design-specific and design-independent functional models together enable the evaluation of a candidate design. the discovery of new design constraints, the reformulation of the design problem, and the incorporation of the modified constraints into the process of case-based design generation.

Another key characteristic of creative design is the use of innovative strategies for adapting a past design to meet the specifications of a new problem. Cross-domain analogical transfer of knowledge is an example of an innovative adaptation strategy. Bhatta and Goel [1993a, 1993b] have shown how design-specific and design-independent models together enable analogical transfer of design knowledge from one engineering domain (e.g., electrical circuits) to another (e.g., heat exchangers). They describe how design-specific functional models enable the learning of design-independent physical processes (e.g., the process of heat flow) and engineering mechanisms (e.g., the cascading mechanism) from specific design experiences in one domain, and how these abstract processes and mechanisms can be used for solving design problems in a different domain.

3 Publications, Presentations and Reports

Invited Talks:

Kolodner, J.L. A Case-Based Approach to Creativity in Problem Solving, Distinguished Lecture at Trinity College, Hartford, Connecticut, April 1993.

> **Abstract:** In case-based reasoning, new problems are solved by remembering (retrieving) previous problem situations similar to a new one and adapting retrieved solutions to fit the new problem. Case-based reasoning is useful for design tasks, planning, diagnosis problems, and common-sense problem solving. It is an inference method people use quite often in their day-to-day reasoning for both expert and common-sense tasks, and it provides an alternate way of building expert systems.
>
> If we take case-based reasoning seriously as a cognitive model of the problem solving people do, then we can use it to begin to explain creative problem solving. A case-based approach to creative problem solving starts with case-based processes at its core and asks how those processes need to be augmented and/or extended and/or redefined so that they can also be used to explain creative thought.
>
> An informal analysis of several instances of creative problem solving has shown us that a major activity creative problem solvers engage in is exploration and evaluation of alternatives, often adapting and merging several possibilities to create a solution to a new problem. I propose a process model of this activity and discuss the requirements it puts on case representations and case-based and other reasoning methods. Some examples from a prototype program will be shown.

Kolodner, J.L. A Case-Based View of Case-Based Reasoning, Invited talk, *AAAI Case-Based Reasoning Workshop*, Washington, D.C., July, 1993.

Kolodner, J.L. Keynote Address: A Case-Based Approach to Creativity in Problem Solving, *First European Workshop on Case-Based Reasoning*, University of Kaiserslautern, Germany, Nov. 1993. J.L. Kolodner will also be presenting invited talks in Holland and Belgium during her trip to Europe in Oct-Nov., 1993.

Kolodner, J.L. Conceptual Foundations of Case-Based Reasoning, two invited talks at GMD and University of Kaiserslautern, Germany, Oct-Nov., 1993.

> **Abstract:** Case-based reasoning has matured in the past several years from a research idea to an approach to building applications and on to providing an approach to addressing research problems that have been otherwise inaccessible. Doing a good job of either of these tasks requires intimate knowledge of CBR's conceptual underpinnings. Unfortunately, the CBR community has done a poor job of articulating these. In particular, there are major misconceptions about indexing and about the role of rules and general knowledge in reasoning. I address those issues, beginning by illustrating the results of these misconceptions, continuing by making clear the approach CBR puts forth as a paradigm, ending by discussing indexing and knowledge issues in some detail.

[Ram 1993]
Ram, A. Creative Conceptual Change, Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society, pp. 17-26, June 1993.

Abstract: Creative conceptual change involves (a) the construction of new concepts and of coherent belief systems, or theories, relating these concepts, and (b) the modification and extrapolation of existing concepts and theories in novel situations. I discuss these and other types of conceptual change, and present computational models of constructive and extrapolative processes in creative conceptual change. The models have been implemented as computer programs in two very different task domains, autonomous robotic navigation and fictional story understanding.

Publications:

[Kolodner 1993bk]
Kolodner, J.L. *Case-Based Reasoning*. Morgan-Kaufman Publishers, Inc., San Mateo, CA, 1993.

[Kolodner 1993a]
Kolodner, J.L. and Wills, L.M. Case-Based Creative Design, *AAAI Spring Symposium on AI and Creativity*. Stanford, CA. March 1993. To be reprinted in an edited book based on the papers presented at the Creativity Symposium and in a special Autumn edition of *AISB Quarterly* on AI and Creativity, edited by Terry Dartnall. (This special edition will contain a select few papers that provide an overview of the field and that give an indication of future directions.)

Abstract: Designers across a variety of domains engage in many of the same creative activities. Since much creativity stems from using old solutions in novel ways, we believe that case-based reasoning can be used to explain many creative design processes.

[Kolodner 1993b]
Kolodner, J.L. and Wills, L.M. Paying Attention to the Right Thing: Issues of Focus in Case-Based Creative Design, *AAAI Case-Based Reasoning Workshop*. Washington, D.C., July 1993.

Abstract: Case-based reasoning can be used to explain many creative design processes, since much creativity stems from using old solutions in novel ways. To understand the role cases play, we conducted an exploratory study of a seven-week student creative design project. This paper discusses the observations we made and the issues that arise in understanding and modeling creative design processes. We found particularly interesting the role of imagery in reminding and in evaluating design options. This included visualization, mental simulation, gesturing, and even sound effects. An important class of issues we repeatedly encounter in our modeling efforts concerns the focus of the designer. (For example, which problem constraints should be reformulated? Which evaluative issues should be raised?) Cases help to address these focus issues.

[Kolodner 1993c]
Kolodner, J.L., et al. Creativity is in the Mind of the Creator: Review of Boden's *The Creative Mind*, submitted to *Behavioral and Brain Sciences*, Princeton, NJ, 1993.

[Moorman 1993]
Moorman, K. and Ram, A. A Functional Theory of Creative Reading, submitted to *Psycgrad Journal*, Oct. 1993.

Abstract: Reading is an area of human cognition which has been studied for decades by psychologists, education researchers, and artificial intelligence researchers. Yet, there still does not exist a theory which accurately describes the complete process. We believe that these past attempts fell short due to an incomplete understanding of the overall task of reading; namely, the complete set of mental tasks a reasoner must perform to read and the mechanisms that carry out these tasks. We present a functional theory of the reading process and argue that it represents a coverage of the task. The theory combines experimental results from psychology, artificial intelligence, education, and linguistics, along with the insights we have gained from our own research. This greater understanding of the mental tasks necessary for reading will enable new natural language understanding systems to be more flexible and more capable than earlier ones. Furthermore, we argue that creativity is a necessary component of the reading process and must be considered in any theory or system attempting to describe it. We present a functional theory of creative reading and a novel knowledge organization scheme that supports the creativity mechanisms. The reading theory is currently being implemented in the ISAAC (Integrated Story Analysis And Creativity) system, a computer system which reads science fiction stories.

[Bhatta 1993a]
Bhatta, S. and Goel, A. Discovery of Physical Principles from Design Experiences. To appear in a Special Issue on Machine Learning in Design of the International Journal *AI in Engineering Design, Analysis, and Manufacturing*, 1993.

Abstract: One method for making analogies is to access and instantiate abstract domain principles, and one method for acquiring knowledge of abstract principles is to discover them from experience. We view generalization over experiences in the absence of any prior knowledge of the target principle as the task of hypothesis formation, a subtask of discovery. Also, we view the use of the hypothesized principles for analogical design as the task of hypothesis testing, another subtask of discovery. In this paper, we focus on discovery of physical principles by generalization over design experiences in the domain of physical devices. Some important issues in generalization from experiences are what to generalize from an experience, how far to generalize, and what methods to use. We represent a reasoner's comprehension of specific designs in the form of structure-behavior-function (SBF) models. An SBF model provides a functional and causal explanation of the working of a device. We represent domain principles as device-independent behavior-function (BF) models. We show that (i) the function of a device determines what to generalize from its SBF model, (ii) the SBF model itself suggests how far to generalize, and (iii) the typology of functions indicates what method to use.

[Bhatta 1993b]
Bhatta, S. and Goel, A. Learning Generic Mechanisms from Experiences for Analogical Reasoning. In *the Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, June 18-21, 1993, Boulder, CO.

Abstract: Humans appear to often solve problems in a new domain by transferring their expertise from a more familiar domain. However, making such cross-domain analogies is hard and often requires abstractions common to the source and target domains.

Recent work in case-based design suggests that generic mechanisms are one type of abstractions used by designers. However, one important yet unexplored issue is where these generic mechanisms come from. We hypothesize that they are acquired incrementally from problem-solving experiences in familiar domains by generalization over patterns of regularity. Three important issues in generalization from experiences are what to generalize from an experience, how far to generalize, and what methods to use. In this paper, we show that mental models in a familiar domain provide the content, and together with the problem-solving context in which learning occurs, also provide the constraints for learning generic mechanisms from design experiences. In particular, we show how the model-based learning method integrated with similarity-based learning addresses the issues in generalization from experiences.

[Bhatta 1992]
Bhatta, S. A Model-Based Approach to Analogical Reasoning and Learning in Design. Technical report GIT-CC-92/60, Ph.D. Proposal, Nov. 1992.

Abstract Analogy is often believed to play an important role in the reasoning underlying innovation and creativity. The ability to make analogies between distant situations or domains (i.e., cross-domain analogies) appears to be crucial for innovation and creativity. However, making cross-domain analogies often involves learning shared abstractions as well as reasoning mediated by the abstractions. We hypothesize that structure-behavior-function (SBF) models at different levels of abstraction provide the right knowledge to facilitate analogical reasoning, ranging from within-domain to cross-domain analogies. We call such analogical reasoning *model-based analogy*.
A mental model is characterized by the types of information it captures such as causal, functional (teleological), and structural relations between the entities in a system or a situation. We represent device-specific models (i.e., models of specific designs) as SBF models and device-independent models (i.e., models of physical principles, processes, and generic mechanisms) as behavior-function (BF) models.
An important issue concerning mental models is their origin. One method for acquiring knowledge of these models is to "discover" them from experience. We hypothesize that SBF models at a lower level of abstraction (e.g., device-specific models) provide both the content and constraints for learning BF models at higher levels of abstraction (e.g., device-independent models) by generalization.
We propose an integrated architecture for design by model-based analogy and for learning of shared abstract models. We are currently implementing the architecture in a system called IDEAL (Integrated "DEsign by Analogy and Learning"). We plan to evaluate it in the context of the design of physical devices, such as heat exchangers and electric circuits.

[Prabhakar 1992]
Prabhakar, S. and Goel, A. Integrating Case-Based and Model-Based Reasoning for Creative Design: Constraint Discovery, Model Revision, and Case Composition. In *Proceedings of the Second International Conference on Computational Models of Creative Design*, Dec. 1992, Heron Island, Australia.

Abstract: Creative Design can be defined as introducing new design variables into the existing design problem space. Many devices fail to perform normally in a new operating environment. This is because the environment imposes new constraints on the device which may not be addressed in the design knowledge. We present a model. Performance-Driven Creativity (PDC). for creative design that introduces new variables into design problem space by discovering and addressing new constraints on the design knowledge. PDC is an extension of KRITIK [Goel. 89] which integrates model-based reasoning and case-based reasoning to come up with creative designs. We have identified three case-bases that help in PDC: (i) Case-base of design experiences that were encountered in the past. (ii) Case-base of previous experiences of failure output behaviors. and (iii) Prototypical behaviors. The knowledge in these cases is modeled using a Structure-Behavior-Function (SBF) model. The PDC task has been decomposed into: (i) Discovery of New Constraints. (ii) Formation of Behaviors for the Constraints. and (iii) Composition of Behaviors to arrive at the final design that satisfies all the constraints identified. In the process of creative design. different models get composed into a single model that represents the final design knowledge. We illustrate our ideas in the design of coffee-maker that can withstand cold environmental conditions.

4 Transitions and DoD Interactions

Because our exploratory study involved a team of students collaborating on a design. it is of considerable interest to researchers studying human-computer collaboration. We are sharing the transcripts and data collected from our exploratory study with researchers at the DEC-Cambridge Research Laboratory who are studying cooperation among heterogeneous agents. In addition. we have been invited to participate in the AAAI-93 Fall Symposium on Human-Computer Collaboration in October.

5 Software and Hardware Prototypes

We are developing a software prototype which integrates a number of primary mechanisms. including capabilities for retrieval, evaluation, adaptation, elaboration (of both solution and specification). and projection of outcomes of proposed alternatives. The prototype has a flexible. opportunistic control structure which allows us to keep focus tactics separate, explicit, and modifiable. This will enable us to explore various strategic control strategies that string together the primary mechanisms. causing complex and interesting interactions from which creative processes emerge.

This experimental system has two memory components: (1) a long-term episodic memory (which includes design cases) and (2) a working memory of the evolving specifications and proposed design alternatives. The working memory organizes the proposed solutions and solution fragments with respect to each other, comparing them along dimensions corresponding to criteria and constraints imposed by the current specification. This memory organization will allow us to explore some interesting working memory issues. such as how it is maintained when it gets large, what things tend to be accessible. what influences accessibility, and how it gets reconstructed when one leaves a problem and then comes back to it.

The design specification that is being evolved by the primary mechanisms is used in two ways. One is as a probe to flexibly retrieve relevant cases. (In case-based reasoning terms, the specification evolution process is one of situation assessment and index transformation.) The other use is as a dynamically changing indexing vocabulary with which to interpret and organize alternatives in working memory. Not only are intentionally proposed solutions recorded in working memory. but also alternatives that are observed in the external environment. This will be used to model the serendipitous recognition of solutions to pending problems as a process of re-interpretation in the context of the current problem.

6 Photographs, Vugrafs or Videotapes

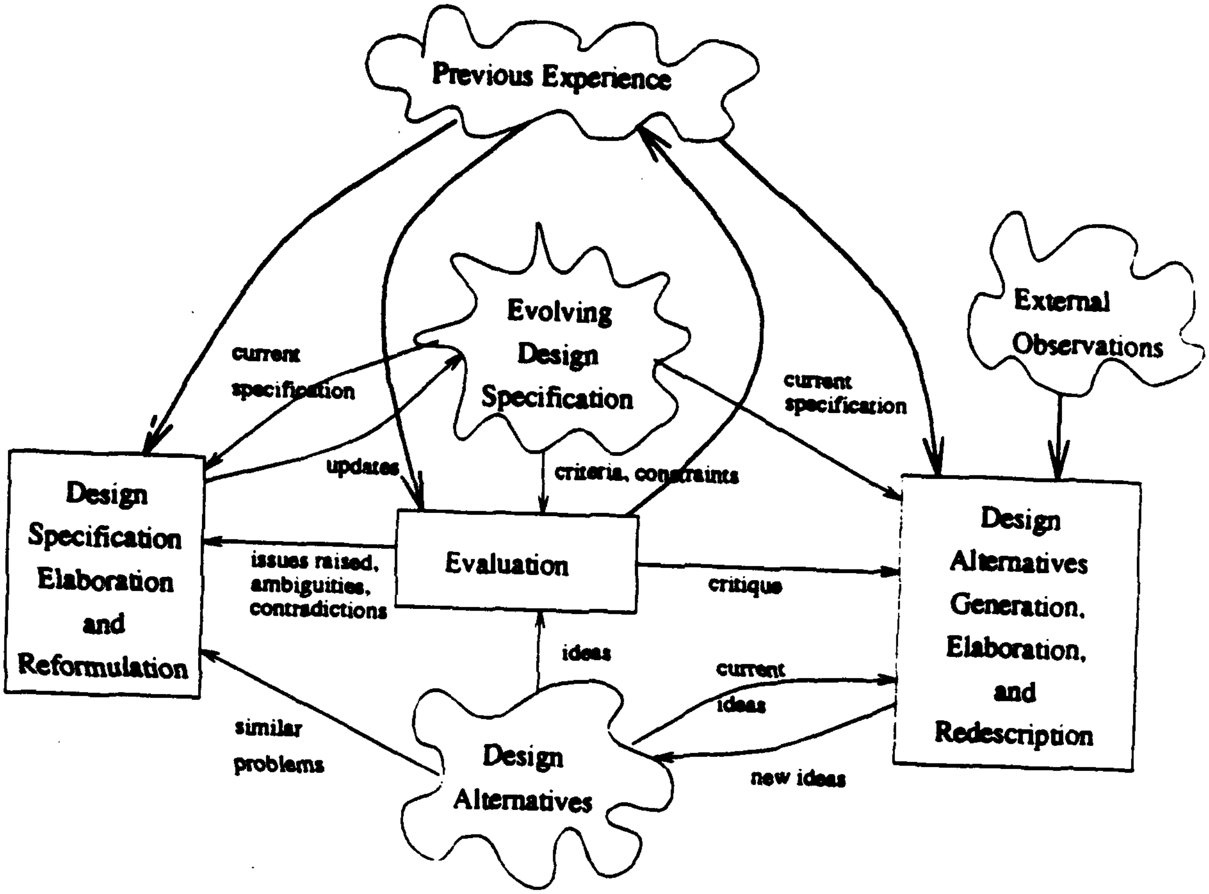


Figure 1: Our model of the creative design process.

CASE-BASED CREATIVE DESIGN

Janet L. Kolodner and Linda M. Wills
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
jlk@cc.gatech.edu, linda@cc.gatech.edu

Abstract. Designers across a variety of domains engage in many of the same creative activities. Since much creativity stems from using old solutions in novel ways, we believe that case-based reasoning can be used to explain many creative design processes.

1. Introduction

Designers across different domains perform many of the same creative activities, whether they are involved in designing artifacts or processes. These activities can be described by contrasting them to routine design activities. In general, routine design repeats old designs in obvious ways, adapting them by well-known and often-applied adaptation strategies. Routine design assumes a completely specified problem is given and little effort is applied to elaborating or designing a feasible specification.

The kind of design we call creative, on the other hand, includes a process of "designing the design specification" (Tong, 1988), going from an incomplete, contradictory, and underconstrained description of what needs to be designed to one with more detail, more concrete specifications, and more clearly defined constraints. Creative design also often includes a process of generating and considering several alternatives, weighing their advantages and disadvantages, and sometimes incorporating pieces of one into another. It involves using well-known design pieces in unusual ways or modifying well-known designs in unusual ways. Creative designers frequently engage in cross-domain transfer of abstract design ideas. They also often recognize alternative uses or functions for common design pieces (e.g., using a styrofoam cup as a boat).

Figure 1 gives a rough sketch of the main processes we hypothesize to be involved in creative design and how they interact with one another. The designer continually updates the design specification as well as a pool of design ideas under consideration. Each alternative generated is evaluated to identify its advantages and disadvantages and to check that it satisfies the constraints in the current design specification. A key part of evaluation is "trying out" the alternative (e.g., through experimentation or mental simulation). This generates a more detailed description of the alternative, including the consequences of its operation and how environmental factors affect it.

Evaluation drives both the updating of the design specification and the modification and merging of design alternatives. It raises questions of legality or desirableness of features[1] of a design alternative and it detects contradictions and ambiguities in the specification. The resolution of these questions, contradictions, and ambiguities serves to refine, augment, and reformulate the design specification. On the generative side, evaluation identifies advantages and disadvantages of alternatives which often suggest interesting adaptations or ways of merging alternatives. Also, sometimes the description of a problem noticed during evaluation can be easily transformed to a description of how its solution would look.

The three processes interact opportunistically. The generative phase, guided by critiques from the evaluation phase, watches for opportunities to merge or adapt design ideas to create new alternatives. The design specification is incrementally updated as ideas are tested and flaws or desirable features become apparent.

The continual elaboration and reformulation of the problem (i.e., the design specification) derives abstract connections between the current problem and similar problems in other domains, facilitating cross-contextual transfer of design ideas. Continual redescription of what the solution (i.e., the evolving design) looks like primes the designer for opportunistic recognition of alternative functions of objects.

This paper describes the nature of these processes and proposes ways of modeling them. Since all three processes rely heavily on previous design experiences, case-based reasoning (Kolodner, 1993) can play a large role in modeling them. Research in case-based reasoning has provided extensive knowledge of how to reuse solutions to old problems in new situations, how to build and search case libraries (for exploration of design alternatives), and how to merge and adapt cases. Many of the activities of creative designers can be modeled by extending routine problem solving processes that exist in current case-based systems.

[1] The features of a design alternative are not only its structural characteristics and physical properties, but also relations between combinations of features.

AAAI-93 Spring Symposium on AI & Creativity, Stanford, CA, March 1993

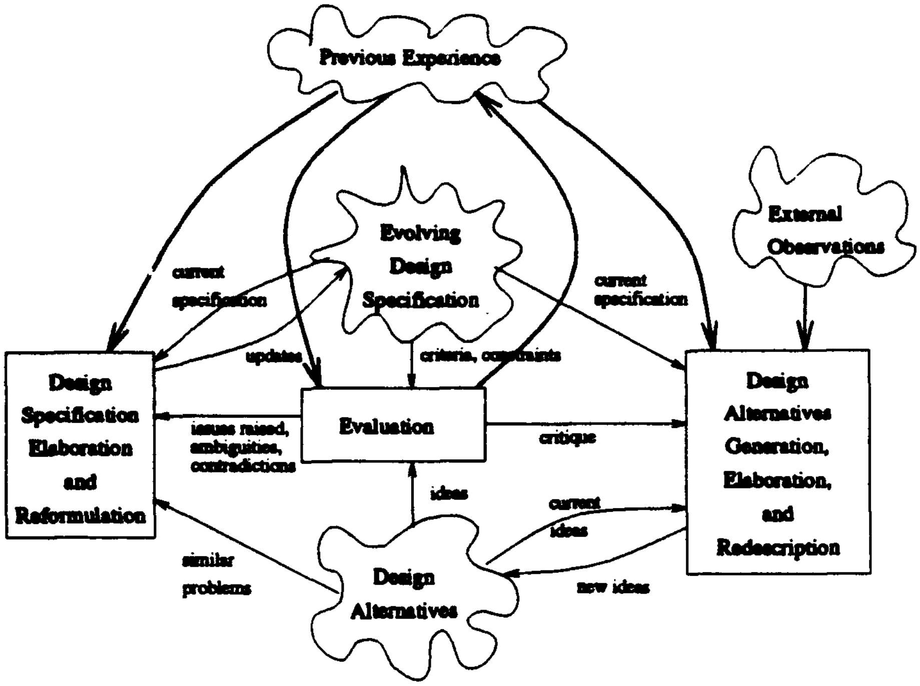


Figure 1: Rough sketch of creative design processes.

We give examples to illustrate these activities, which we have collected in studying the problem solving reports and protocols of designers in a variety of design disciplines. These include software design, meal planning, science lesson planning, architectural design, and mechanical design. Many of the anecdotes related in this paper come from an exploratory study we conducted of a student mechanical engineering (ME) design project. The design task was to build a device to quickly and safely transport as many eggs as possible from one location to another. The device could be constructed from any material, but had to satisfy a set of size, weight and cost restrictions. The initial description of the problem was vague, ambiguous, and incomplete, requiring a great deal of elaboration and reformulation. One of us participated in the seven-week project as a member of a four-person team. Active participation in the project allowed us to become immersed in the issues the students were dealing with and to openly converse with the students at all stages of the design as a useful team member, rather than as an outside observer. This led to many of the insights described in this paper.

2. Specification Refinement

Design specifications are rarely well-defined. In general, they are incomplete, leave many different ways to solve a problem, and are often unnecessarily overconstrained. An important part of design is redefining the design specification. This includes elaborating the constraints and criteria the design should satisfy and extensively restructuring the problem (Goel and Pirolli, 1989).

2.1 ELABORATION

In general, a designer has goals and guidelines that are not in the initial design specification itself but whose violation or achievement can be noticed. For example, a meal planner might like meals to be easy to prepare, but may not include this in every design specification. Goel and Pirolli (1989) identify several classes of constraints that are of this nature, including domain-specific technical constraints (such as structural soundness), legislative constraints (such as building codes), common sense, pragmatic constraints (for example, "short construction time" or personal safety), and self-imposed, personal preferences (such as "not spicy").

Elaboration involves making these constraints and criteria explicit, consistent, complete, and unambiguous. We hypothesize that this is driven in part by the process of evaluating each alternative generated so far. Evaluation drives elaboration by satisfaction or dissatisfaction with an alternative and by an inability to evaluate. Elaboration is also driven by an inability to generate satisficing alternatives, and by opportunity. These are discussed in this section.

Many design alternatives arise from remembering or looking for solutions to old design problems. Such design cases are, in general, similar to the new situation on important dimensions, but are more complete. Additional aspects fuel elaboration by bringing up new constraints or criteria to consider. They are evaluated for applicability to the current design problem. The results are used to update the design specification: if the case is applicable, more detailed constraints are added; if the case is rejected, constraints are added to prohibit the aspects that were unacceptable.

For example, while designing a manufacturing research center on the Georgia Tech campus, Terry Sargent visited existing manufacturing centers and precision engineering laboratories around the world. Examining these options helped him decide what criteria and constraints were important and how to prioritize them. One technical center he visited has flexible utilities which can be tapped into at any location in the building (e.g., an air duct can be added anywhere). On the other hand, all of its research laboratories are internal and the building is too dark. From his examination, Sargent formed a wishlist of constraints for his building to satisfy, including having flexible utilities, external offices, and letting in plenty of sun.

This illustrates two of the ways evaluation drives elaboration: by satisfaction and by dissatisfaction with an alternative. A third way is by an inability to evaluate. This occurs when there is a lack of information in the specification to confirm or reject the legality and desirableness of features. It suggests new constraints and criteria to add or existing constraints to disambiguate.

An example arose in the ME design project, where a possible starting location of the device was from the center of a wading pool of water. The team discussed the idea of launching an egg-carrying device from a model battleship. To determine whether this was legal, they needed to know whether it was all right to leave parts of the device behind as it operated. The answer to this question added to the problem description.

Elaboration is also driven by an inability to generate satisficing alternatives. In general, this results in relaxing constraints (i.e., making a compromise). In the ME project, the students originally wanted to carry more than than a dozen eggs, but could think of no design ideas that would allow a large number of eggs to be carried safely, given the amount of protective cushioning required and the space restrictions. This led the students to relax their preference for the device to have a high egg-carrying capacity.

Finally, elaboration is driven by opportunity. If the evaluation process is aware of the designer's other goals, it can be opportunistic. For example, a meal planner whose immediate goals were to use leftover rice for dinner remembered a breakfast dish. Since she needed to eat breakfast too, she decided to relax the dinner goal and use the rice for breakfast. This required reasoning about priorities and alternative ways of doing things. If rice is the only thing of substance available for dinner, then using the rice for tomorrow's breakfast is a poor idea. If, on the other hand, there are plenty of other things available for dinner and/or the eater didn't really want to eat rice anyway, then using it for breakfast solves two problems. So, evaluation may allow a reasoner to opportunistically realize that a solution is good, even though it does not fit the design specification. This can lead to a change in the relative importance of goals and constraints in the current problem description.

2.2 REFORMULATION

Another major activity in designing the design specification is reformulating the problem – redescribing the problem so that the solution is easier to find. There are several ways alternative views of a problem can be generated.

One way stems from making a design alternative more concrete, e.g., by mentally visualizing it or acting it out. The more detailed description of the solution sometimes suggests a new description of the problem. For example, in the ME design project, while considering how to move eggs out of a pool of water, one student made an analogy to submarines launching missiles. He acted out the launch with his pen as he spoke. His description reminded another student that submarines launch missiles one at a time. This led to reformulating the problem from one of moving all eggs as a group to moving eggs individually.

It is an open question exactly how a more detailed description of a solution can suggest a reformulation. It may be that the visualization of the submarine launching is making assumptions explicit. It is challenging constraints that have been inherited from previously considered options, but which are not essential, e.g., the constraint on how many eggs should move at once.

Another problem reformulation technique is to explore and stretch the problem constraints and exploit any loop-holes found. For example, a designer trying to "design a building between two buildings" (Goel and Pirolli, 1989) might ask how close the middle building can be to the two adjacent buildings. By taking closeness to the limit, the designer can reformulate the problem as "connect two buildings together."

Finally, a third way an alternative view of a problem sometimes arises is from realizing part of a solution and then reducing the problem to making that happen. For example, Maier (1931) describes an experiment in which subjects were given the problem of connecting together two strings that hung vertically a large enough distance apart that the person could not hold one string and reach the other. The solutions depend on describing the problem in different ways: "how to make one string longer," "how to make one string stay in middle and bring the other string to it," "how to extend my reach to pull one string to the other," and "how to make one string move to the other." Maier showed that subtly giving the hint of making one string sway often helped the subjects come up with the fourth reformulation (which led to the solution of tying a weight to the string, making it swing like a pendulum toward the other string).

Turner (1991) provides an initial attempt to model the problem reformulation process, which he implemented in a program called MINSTREL. Turner proposes a case-based model of creative reasoning in which a given problem is transformed into a slightly different problem and then used as a probe to a case library. A recalled solution to the new problem is then adapted back to the original problem (using solution adaptations that are associated with the problem transformations). A set of "creativity heuristics" is used to transform the problem. Examples include generalizing a constraint (and perhaps suspending it altogether), and adapting a constraint to require a related, but slightly different outcome (e.g., injuring instead of killing).

Unfortunately, MINSTREL does not address important focus of control problems. For example, what guides the problem reformulation? Which features or constraints should be varied? Figuring out what to change and how seems to be a major part of recasting problems. We believe that incorporating feedback from the evaluation of proposed alternatives can provide focus.

3. Idea Exploration

Generating design alternatives is an incremental, opportunistic process that is tightly interleaved with specification refinement and evaluation. Three primary ways in which ideas are put on the table for consideration are retrieval of previous design experiences, recognition of current experiences or design pieces in the current environment as potential solutions, and modifying or combining existing options to produce new ones.

3.1 REMINDING

An expert designer knows of many design experiences, accumulated from personally designing artifacts, being given case studies of designs in school, and observing artifacts designed by others. Our observations and analyses lead us to propose that reminding of these experiences is crucial to generating design alternatives. When a design experience is recalled, it suggests a potential solution that can be critiqued with respect to the new problem, adapted to meet the needs of the new situation, or merged with other proposed solutions.

Designers frequently choose an already well-known framework (or generic case) for a problem and then fill it in. Reusing solution structures in this way allows designers to avoid recomputing useful compositions of design pieces. We call this process "framing a solution." The framework provides the glue holding the pieces of the design together. The creativity comes in filling in details and in dealing with inconsistencies when merging alternative pieces.

Such framing occurs in domains, such as bridge design and engine design, where well-known frameworks exist and where constraints holding the pieces of problems together are quite complex. In software engineering, frameworks exist as widely-used computational fragments, called clichés (Rich and Waters, 1990). Johnson and Foote (1988) have defined a similar notion of "frameworks" for reuse of object-oriented software. In other domains, such as architectural design, creating the framework is a primary piece of the creative process. This involves deciding which aspects of a problem specification are most important to deal with first and inferring structural aspects of a solution from them.

We hypothesize that unorthodox design alternatives tend to come from non-obvious remindings. Some are based on abstract similarities, resulting in cross-contextual remindings. Other remindings are based on derived or computed features rather than available ones.

If reminding is so important to generating alternatives and if it requires derived or abstract features, we must determine which kinds of derived features tend to be most useful for design, whether there is a set of derived features that is common to design across domains, and when those features get derived.

In her investigation of story writing, Dehn (1989) stresses the importance of reusing old ideas in new ways. Of particular importance is having processes that are able to generate multiple alternatives for several parts of a problem and put them together in unusual ways. This requires processes that can search memory for things that might be represented in a way that is different from the representation of the current problem. Old cases must be seen in a new light.

Recent studies of creative problem solving protocols (Kolodner and Penberthy, 1990) suggest that anticipatory indexing is not sufficient to fully explain retrieval. Features that were not salient at the time a case was experienced might be important for retrieval in the current situation. Drawing new, abstract connections might be a result of re-indexing cases in terms of what is now relevant or important. We hypothesize that by continually updating the design specification, designers derive abstract connections between the current problem and similar problems (possibly in other domains). These abstractions can be used to see previous cases differently.

Selfridge (1990) claims that people tell stories to re-index them under new generalizations that have been learned since the story was first acquired. A key open question he identified is how does a person know what stories to tell? One possibility is that they are the ones the person is reminded of or has experienced recently. The person may have been reminded of them through a different set of features than the generalized features they are re-indexed under. While working on a design problem, designers often perform sensitized recognition of current design options and objects in their environment and they continually re-examine and re-index all ideas recently brought up or experienced. This is discussed further in the next section.

Retrieval can be automatic or strategic (i.e., based on intentional elaboration strategies that help jog a designer's memory). Strategic retrieval is promoted by design team communication. Team members describe abstract ideas to each other in terms of concrete examples, analogies, and metaphors. Trying to recall an appropriate example often involves applying elaboration strategies to an index. For example, the person might reflect on "where have I seen something like this before?" and "in what situations might I have seen something like this?" This often results in identifying opportunities to reuse existing objects or devices in the current design.

Team communication plays an additional role in idea generation: ambiguity in communication is generative. In general, when working together, team members try to recognize and understand each others' ideas, plans, and goals from their actions, words, and sketches. Sometimes there is ambiguity in the interpretations which often helps generate more ideas (increases fluidity of concepts) and can lead to function sharing optimizations. Goel (1992) studied the generative role ambiguity plays in informal sketching. In our informal study, we have noticed that interaction among multiple designers amplifies its effects.

3.2 SENSITIZED RECOGNITION

As designers become deeply involved in design problems, they start to recognize objects in their environment as solutions to parts of the design problem. Often the objects are seen as having alternative, unusual functions or uses.

For example, in the ME design project, the students were considering using a spring launching device and went to a Home Depot (a home improvement store) to look into materials. While comparing the strengths of several springs by compressing them, they noticed that the springs bent. One student mentioned that if they were to use springs, they would have to encase the springs in collapsible tubes to prevent bending. Later, as they walked through the bathroom section of the store, they saw a display of toilet paper holders. They immediately recognized them as collapsible tubes that could be used to support the springs.

The key to sensitized recognition is refining the description of the solution. The process of critiquing proposed ideas often yields descriptions of what an improved solution would look like: what properties it would have, what function it should provide, and what criteria it satisfies. This primes the designer to opportunistically recognize possible solutions in observations of the external world and in recently considered design options.

3.3 ADAPTATION

Previous work has looked at adapting old solutions to fit new problems. In creative design, it sometimes makes sense, in addition, to adapt one's goals to fit an old solution rather than changing the old solution to fit the new problem (e.g., using rice for breakfast rather than dinner). Previous work (Hinrichs 1992) has looked at routine adaptation strategies (e.g., deletion, addition, substitution) but not at use of "off-the-cuff" ones (i.e., those developed in response to a particular problem). Some of these arise from examining a causal model, some from adapting well-known adaptation strategies, and some from applying well-known adaptation strategies in novel ways. For example, novelty can result from substituting something different than the usual thing or from relaxing well-known structural constraints.

3.4 MERGING

In routine design, parts of several designs are often merged, but in general, the parts are non-overlapping (e.g., dessert from one meal might be used with a main dish from another meal). In more novel design, several suggestions for solving the same part of a problem might be merged to come up with a solution (e.g., in deciding to have salmon fettuccine and salad for dinner, a meal planner might have remembered three previous cases, a meal with fish, a one-dish meal and a pasta meal, and merged desirable features from each).

Merging pieces of several solutions into one design is relatively simple if the pieces are consistent with each other. Either a previous case will suggest a way of combining them, an adaptation heuristic will know how or combination will be obvious. Merging is more complex when the pieces are not obviously consistent. We have two hypotheses about how creative merging of several alternative solutions might work. First, some adaptation heuristics might exist that can provide general guidelines and suggestions for non-routine merging. Second, cases from other domains may provide guidelines and suggestions for non-routine merging. The challenges here are to find the adaptation heuristics and to discover the descriptive vocabulary that allow cross-contextual remindings of the appropriate kinds to take place.

3.5 FUNCTION SHARING

Often function-sharing optimizations arise from merging within the same design. This occurs when an existing part of the design can be seen to fulfill another purpose. (This is a special case of sensitized recognition.) An interesting form of this type of merging occurred in the ME design project. The students had decided to use a cylinder to carry the eggs. One student related an episode from the children's science TV show Beakman's World that had caught her eye as she was flipping through channels. The episode showed how to make a coffee can that rolled back to you when you rolled it away. It attached batteries as weights to rubberbands, strung through the center of the can. The weights caused the rubberbands to get wound up as the can rolled. As the rubberbands unwound, they caused the can to roll back to the starting location. The students discussed whether this could be modified for use in their design (e.g., wind the rubberbands up and let their unwinding launch the device). They criticized the rubberband and battery part for taking up too much space and for adding too much weight, since the task had strict space and weight restrictions. One student then suggested the interesting optimization of letting the eggs themselves be used as the weights. This alleviated both the space and the weight problem. One aspect that was non-routine about this is that the student looked beyond the structure of the device to its cargo to find what to share.

4. Evaluation

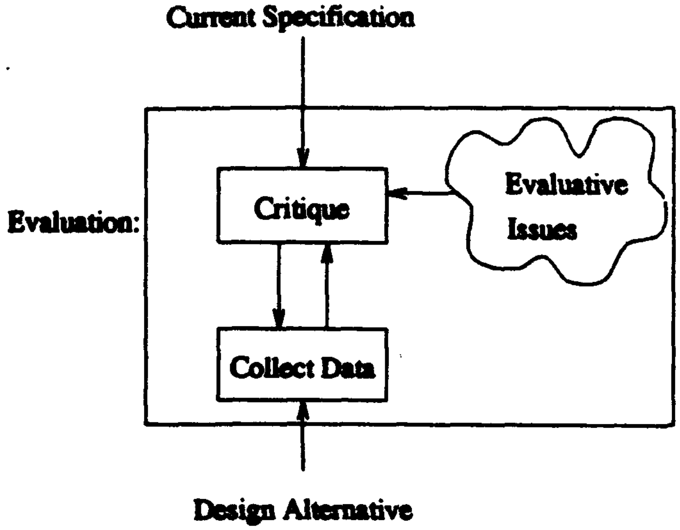


Figure 2: Processes involved in evaluation.

The evaluation process checks each design option that is generated against the current design specification. It forms a critique, identifying how well the option satisfies the constraints or how badly it fails. It also notices questionable features whose desirableness or legality are unknown. In addition, it raises evaluative issues and guidelines that are not found in the current specification, but which are based on the designer's experience. Some of these are always raised. For example, in algorithm design, issues of correctness, completeness, and time and space efficiency are routinely considered. Others (e.g., elegance) are recalled or derived based on features of the alternatives examined.

This information is used by both the specification refinement process (elaborate and reformulate) and the idea exploration process (generate, elaborate, redescribe). The issues raised point out opportunities to augment or refine the design specification. The pros and cons that are described in the critique of a design idea are used by the idea exploration process to compare the idea to other options, merge and adapt alternatives, and improve promising ideas.

We view evaluation as consisting of two interacting processes, as shown in Figure 2. One process critiques the design alternative on the basis of the current specification and the evaluative issues. The features examined in this critique are not only the structural characteristics of the design artifact, but also information about how it behaves, the consequences of its operation, and how environmental factors affect it. The second process collects this information by performing simulations and experimentation.

In the ME design project, the students often mentally simulated proposed options and checked the results. For example, when the idea of launching each egg individually rather than as a group was considered, the students imagined that the eggs would all land on top of each other which could cause breakage and an unstable target spot. Identifying this problem through mental simulation led to an adaptation of the proposed solution which was to rotate the launch mechanism so that the eggs would each land in a different location.

In addition to simulating the proposed option in the general case, designers also propose hypothetical situations to simulate. For example, the ME students asked, "What if it is raining on the day of the competition?" and "What if the terrain the device must traverse is rough or steep?" Simulations of hypothetical situations test the robustness of the solution. The hypothetical situations pertain to all phases of a design artifact's lifecycle, including its construction and maintenance, as well as its use. For example, a designer might try to imagine someone trying to repair some part of the design that is vulnerable to failure and consider whether the part is accessible for maintenance.

Concrete experimentation of design alternatives is a valuable way of collecting data. Some aspects and outcomes of an option only become apparent through real-world testing. For example, during the ME design project, the students tested the ability of potting sponge (used in floral displays) to cushion eggs. When an egg was placed in it and dropped, the sponge compressed to a powder, decreasing its protective ability and reusability. This led the students to search for a material that did not permanently compress and was reusable.

Some simulations or experiments might be proposed by the critiquing process when it requires additional information about the design option to judge its strengths and weaknesses. Some hypothetical situations used in simulation might be associated with evaluative issues raised in critiquing the design option.

4.1 EVALUATIVE ISSUES

While critiquing a given design option, a designer considers general evaluative issues that the designer's experience recommends looking into, in addition to how well the option fits the current design specification. There are at least three classes of evaluative issues that designers routinely raise (Kolodner and Penberthy, 1990).

One is *function-directed*. For example, the purpose of recipe creation is to create something that can be eaten, so some questions arise from the concept of edibility. These focus on the taste and appeal of a dish to see if it is edible.

Another class is *derivation-driven*: previous solutions provide a rich and important source of issues if the considerations taken into account in creating them are saved. Consider, for example, the task of trying to decide if tofu can be substituted for cheese in tomato tart. One way the right evaluative issues can be derived is by recalling another case where tofu was to be substituted for cheese. Concerns in that case are likely to be concerns in the current one, too. For example, if in the previous case, the texture of tofu was compared with the texture of the original ingredient, the reasoner might then ask about texture in the current case.

Finally, some questions are *outcome-related*. Previous design cases can be used to project or derive the outcome of the current one. For example, as part of the ME design project, a proposed launch mechanism was considered that consisted of a plastic fish tank base and two toilet paper holders (which provided a spring mechanism). The two holders were attached to the base via plastic prongs protruding from one side of the base. One of the students was concerned that the prongs were vulnerable to breaking, particularly if the springs inside the holders were replaced with stronger springs. She recalled similar plastic prongs had held a protective covering on her stereo speakers, but they had broken off of one speaker when it fell at an angle. The proposed design option was used as a probe to memory to see if instances are already known of it or a similar solution failing. By recalling the stereo speaker case, the students raised the question of whether the proposed design was vulnerable in the same way. It also suggested a hypothetical situation in which to simulate the proposed design: what happens if we provide a large side-ways force to the prongs? Thinking about how this could arise led the students to think about what would happen if stronger springs were required.

Case-based projection can bring up outcome-related issues relevant to any phase of a design's lifecycle, besides its normal use, including its construction and maintenance phases. For instance, one of the buildings Terry Sargent examined when designing the Georgia Tech manufacturing research center was the Pompadeau center, which has all of its mechanical systems showing. He wanted to borrow this idea for its symbolism, but in talking with the managers of the center, he found out that this feature made it difficult to maintain the building. This led him to question whether the same maintenance problems will come up in his design.

5. Discussion

Creative designers operate in a rich context of ideas, some recalled from previous experiences, some recognized in the current external environment, and some generated from adapting or putting together recently considered ideas. An important part of this rich context is concreteness. Details fuel evaluation, which is central to elaborating and redescribing both the problem and the solution. These come from reasoning about specific design cases, which include many additional details besides those aspects that originally brought the case to mind. They also come from experimentation, testing, visualizing, and simulating the solutions.

This suggests three important ways to assist creative design. One is by placing the designer in a rich environment containing concrete design artifacts or detailed descriptions and simulations of existing design artifacts. Another is by facilitating evaluative procedures and proposing hypothetical situations covering the artifact's entire lifecycle. The third is by assisting the designer in reformulating and redescribing what is needed, what constraints or criteria need to be satisfied, and what the solution would look like.

5.1 OPEN CONTROL ISSUES

Our exploratory studies of designers have given us insights into the primary activities involved in creative design. However, many open issues remain. Most center around the underlying control of the various processes and their interactions.

Specification refinement. A key activity in designing the design specification is incrementally bringing evaluation criteria and new problem constraints into focus. This is largely driven by evaluation. An open question is how does noticing a feature of a design option that is either satisfactory, undesirable, or whose status is unknown (due to failure to evaluate) lead to an elaboration of the current specification? One possibility is that it can be guided by the mechanism that detected the questionable feature. For example, one way to detect a problem in a proposed solution is by case-based projection: recalling a failure in a similar solution. This previous case might provide suggestions for fixing the current problem specification. Failure to determine the legality of a feature could point to augmentations to the specification that would push the confirmation or rejection through to completion.

Another important question is: during problem reformulation, how is the designer's attention drawn to particular constraints to explore and stretch? There seems to be give and take between reformulation and evaluation. Evaluation can home in on what is ambiguous or vague in the problem specification and try to take advantage of new views that result from relaxing or pushing the limits of the constraints. Also, when the need to compromise arises, conflicting constraints come into focus and the designer considers how they can be varied. On the other hand, reformulation of the specification can provide additional or improved evaluative measures to strengthen evaluation.

Idea exploration. The critique of proposed solutions guides idea exploration. Of several solutions under consideration, one might be more appropriate than the others or several might each contribute to a solution. Evaluative procedures must be able to evaluate each individual alternative by itself as well as in light of the others. Several open questions arise: How is relative importance among the criteria decided? How are preferences among alternatives made? How does weighing advantages and disadvantages suggest useful adaptations and mergings?

Recalled cases seem to be important here. They suggest solutions, frameworks, design strategies and design philosophies, which can provide constraints with which to evaluate a solution and the preference criteria with which to prioritize the constraints. This also facilitates reformulating the specification, making trade-offs, and relaxing constraints. There may also be general and domain-specific strategies for setting priorities that we haven't discovered yet.

Evaluation. An important and open question is how does the evaluation process know which aspects of a design alternative to focus on? Of all the data collected during simulation and experimentation, which subset is interesting? For example, which data is likely to suggest updates to the design specification or adaptations that lead to new ideas?

Evaluative issues that designers always raise tend to focus on particular features. At the same time, some features seem to draw attention to particular evaluative issues that might not have been considered otherwise. Some of the features are more distinctive or odd and these seem to index directly into the set of implicit criteria held by the designer. An example arose in the ME design project. While testing how well various types of spongy material cushioned eggs when dropped from two stories, a person walked by who had done a design project which also involved protecting an egg from breaking on impact. He said he wrapped the egg in a sponge soaked in motor oil and then stuffed it in a Pringles can (a narrow cardboard cylinder in which potato chips are stacked). One of the aspects that was new about this case, compared to the ideas the students had been considering is the idea of soaking the sponge in motor oil. Focusing on the motor oil aspect reminded the students of their personal preference that the device be clean. The motor oil aspect seemed to index directly into the cleanliness criterion.

Overall Control. Other open questions pertain to how designers decide when to expend effort in one process versus another. For example, when should quick adaptations of existing solution ideas be tried and when should the designer step back and reformulate the problem. One observation we made in the ME student design project was that when flaws were noticed, the students usually preferred to redescribe the solution rather than elaborate or reformulate the problem specification. The students described what was needed in terms of how the structure of the device should be modified to fix the problem (e.g., "the launch mechanism must rotate" or "the springs should be in a collapsible tube") as opposed to describing what function or behavior is desired (e.g., "the eggs should each land at different target locations" or "provide side-to-side support to springs"). The students usually tried to adapt the offending feature, before reformulating the problem. Only when quick adaptations to the solution were not sufficient did they step back, look at the essential problem constraints these specific structural solutions were solving, and then reformulate the problem or find other solutions that could also satisfy these constraints. This is reasonable, since it is cheaper to make small changes to an evolving design solution than to completely reformulate the problem. We need to look for additional types of heuristics people use to control their reasoning processes.

Acknowledgments. We would like to thank the members of the mechanical engineering design team, Otto Baskin, Jon Howard, and Malisa Saratinoranont, for their invaluable cooperation. We thank Farrokh Mistree for allowing us to participate in his mechanical engineering project course and for many thought-provoking discussions throughout. We also appreciate the insightful discussions we have had with Sambasiva Bhatta, Ashok Goel, Colin Potts, Mimi Recker, Spencer Rugaber, and Terry Sargent. This research was funded in part by NSF Grant No. IRI-8921256 and in part by ONR Grant No. N00014-92-J-1234.

References

Dehn, N.J.: 1989, *Computer Story-Writing: The Role of Reconstructive and Dynamic Memory*, Ph.D. Thesis. Yale University.

Goel, V.: 1992, 'Ill-Structured Representations' for Ill-Structure Problems. *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, pp. 130-135.

Goel, V. and Pirolli, P.: 1989, Motivating the Notion of Generic Design within Information Processing Theory: The Design Problem Space. *AI Magazine*, Vol. 10, No. 1, pp. 18-36.

Hinrichs, T.R.: 1992, *Problem Solving in Open Worlds: A Case Study in Design*. Lawrence-Erlbaum Associates, Hillsdale, NJ.

Johnson, R.E. and Foote, B.: 1988, Designing Reusable Classes. *Journal of Object-Oriented Programming*, Vol. 1, No. 2, pp. 22-35.

Kolodner, J.L.: 1993, *Case-Based Reasoning*. Morgan-Kaufman Publishers, Inc., San Mateo, CA.

Kolodner, J.L. and Penberthy, T.L.: 1990, A Case-Based Approach to Creativity in Problem Solving. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, Cambridge, MA. August.

Maier, N.R.F.: 1931, Reasoning in humans II: The Solution of a problem and its appearance in consciousness. *Journal of Comparative Psychology*, vol. 12, pp. 181-94.

Rich, C. and Waters R.C.: 1990, *The Programmer's Apprentice*. Addison-Wesley, Reading, MA.

Selfridge, M.: 1990, Why Do Adults Tell Stories? Why Do Children Play? For the Same Reason: Re-Indexing Old Cases Under New Generalizations. *AAAI Spring Symposium on Case-Based Reasoning*, pp. 72-74.

Tong, C.H.: 1988, *Knowledge-Based Circuit Design*. Ph.D. Thesis. Rutgers Technical Report LCSR-TR-108. Laboratory for Computer Science Research. Hill Center for the Mathematical Sciences Busch Campus, Rutgers University. May.

Turner, S. R.: 1991, A Case-Based Model of Creativity, *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, Chicago, pp. 933-937.

Paying Attention to the Right Thing: Issues of Focus in Case-Based Creative Design

Janet L. Kolodner and Linda M. Wills
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
jlk@cc.gatech.edu, linda@cc.gatech.edu

Abstract

Case-based reasoning can be used to explain many creative design processes, since much creativity stems from using old solutions in novel ways. To understand the role cases play, we conducted an exploratory study of a seven-week student creative design project. This paper discusses the observations we made and the issues that arise in understanding and modeling creative design processes. We found particularly interesting the role of imagery in reminding and in evaluating design options. This included visualization, mental simulation, gesturing, and even sound effects. An important class of issues we repeatedly encounter in our modeling efforts concerns the focus of the designer. (For example, which problem constraints should be reformulated? Which evaluative issues should be raised?) Cases help to address these focus issues.

Introduction

Designers across different domains perform many of the same creative activities, whether they are involved in designing artifacts or processes. These activities can be described by contrasting them to routine design activities. In general, routine design repeats old designs in obvious ways, adapting them by well-known and often-applied adaptation strategies. Routine design assumes a completely specified problem is given and little effort is applied to elaborating or designing a feasible specification.

The kind of design we call creative, on the other hand, includes a process of "designing the design specification" (Tong, 1988), going from an incomplete, contradictory, and underconstrained description of what needs to be designed to one with more detail, more concrete specifications, and more clearly defined constraints. Creative design also often includes a process of generating and considering several alternatives, weighing their advantages and disadvantages, and sometimes incorporating pieces of one into another. It involves using well-known design pieces in unusual ways or modifying well-known designs in unusual ways. Creative designers frequently engage in cross-domain transfer of abstract design ideas. They also often recognize alternative uses or functions for common design pieces (e.g., using a styrofoam cup as a boat).

Figure 1 gives a rough sketch of the main processes we hypothesize to be involved in creative design and how they interact with one another. The designer continually updates the design specification as well as a pool of design ideas under consideration. Each alternative generated is evaluated to identify its advantages and disadvantages and to check that it satisfies the constraints in the current design specification. A key part of evaluation is "trying out" the alternative (e.g., through experimentation or mental simulation). This generates a more detailed description of the alternative, including the consequences of its operation and how environmental factors affect it.

Evaluation drives both the updating of the design specification and the modification and merging of design alternatives. It raises questions of legality or desirableness of features[1] of a design alternative and it detects contradictions and ambiguities in the specification. The resolution of these questions, contradictions, and ambiguities serves to refine, augment, and reformulate the design specification. On the generative side, evaluation identifies advantages and disadvantages of alternatives which often suggest interesting adaptations or ways of merging alternatives. Also, sometimes the description of a problem noticed during evaluation can be easily transformed to a description of how its solution would look.

The three processes interact opportunistically. The generative phase, guided by critiques from the evaluation phase, watches for opportunities to merge or adapt design ideas to create new alternatives. The design specification is incrementally updated as ideas are tested and flaws or desirable features become apparent.

The continual elaboration and reformulation of the problem (i.e., the design specification) derives ab-

[1] The features of a design alternative are not only its structural characteristics and physical properties, but also relations between combinations of features.

AAAI-93 Workshop on CBR, Washington, D.C. July 1993

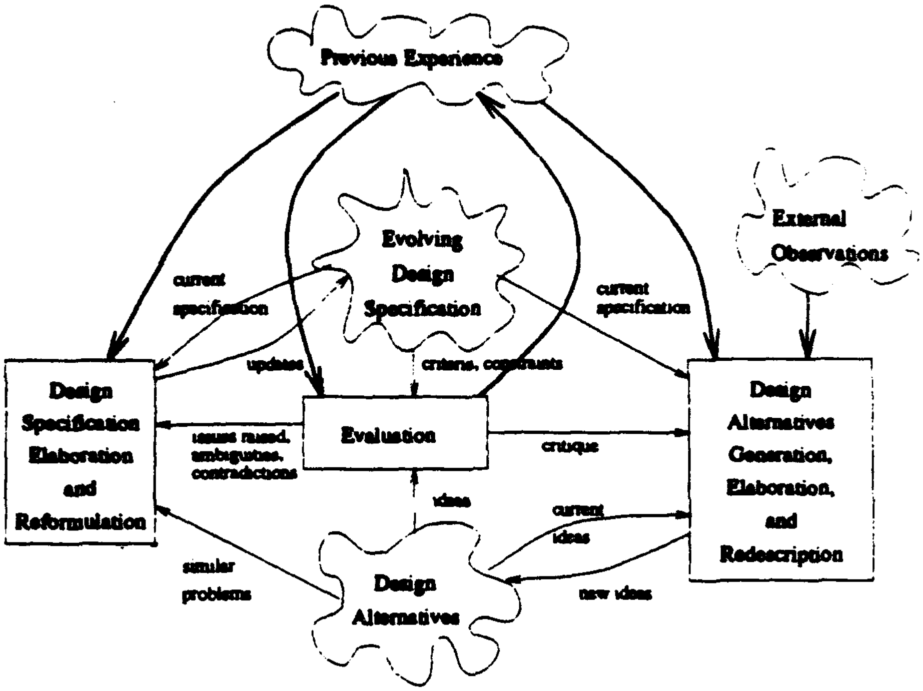


Figure 1: Rough sketch of creative design processes.

stract connections between the current problem and similar problems in other domains, facilitating cross-contextual transfer of design ideas. Continual redescription of what the solution (i.e., the evolving design) looks like primes the designer for opportunistic recognition of alternative functions of objects.

These processes rely heavily on previous design experiences and knowledge of designed artifacts. An expert designer knows of many design experiences. accumulated from personally designing artifacts. being given case studies of designs in school, and observing artifacts designed by others. Our observations and analyses lead us to propose that reminding of these experiences is crucial to generating design alternatives. When a design experience is recalled, it suggests a potential solution that can be critiqued with respect to the new problem, adapted to meet the needs of the new situation, or merged with other proposed solutions.

We believe that case-based reasoning (Kolodner, 1993) can play a large role in modeling these processes. Research in case-based reasoning has provided extensive knowledge of how to reuse solutions to old problems in new situations, how to build and search case libraries (for exploration of design alternatives), and how to merge and adapt cases. Many of the activities of creative designers can be modeled by extending routine problem solving processes that exist in current case-based systems.

Design cases provide a rich collection of details that are used in several ways in addition to generating ideas, including

- reformulating and elaborating the problem specification or proposed solutions,
- predicting the outccme of making certain design decisions,
- enabling visualization and simulation of proposed designs, and
- communicating abstract ideas in concrete terms.

What cases seem to do is to help the reasoner determine how to productively continue reasoning. The question we ask is how? How does the designer know which details to pay attention to? Which aspects of an old design can suggest problem reformulations or can fill in missing details of the specification? During problem reformulation, which constraints should be relaxed or strengthened? How are evaluative questions and criteria incrementally raised to critique the proposed design options?

We call this problem "focus." These issues are relevant in understanding what knowledge must be captured in case libraries, the form this knowledge should be in, and what types of indices are needed to allow retrieval of relevant cases. At the same time, cases help address many of these focus-related issues, particularly raising evaluation criteria and suggesting interesting, useful problem reformulations.

Example Design Episode

We concentrate primarily on an example design episode from an exploratory study we conducted of a student mechanical engineering (ME) design project.

The design task was to build a device to quickly and safely transport as many eggs as possible from one location to another. The device could be constructed from any material, but had to satisfy a set of size, weight and cost restrictions. The initial description of the problem was vague, ambiguous, and incomplete, requiring a great deal of elaboration and reformulation. One of us participated in the seven-week project as a member of a four-person team, rather than as an outside observer. Active participation in the project allowed us to become immersed in the issues the students were dealing with and to observe a great deal of the design process, including "official" as well as informal team meetings (e.g., while choosing materials at a store or while attending class).

The following is a short excerpt from a discussion early in the project concerning how to launch the eggs from the center of a child's wading pool. This excerpt was chosen because it involves a reformulation of the original problem statement. It illustrates the types of design experiences and artifacts the students typically recalled and the variety of ways they used these remindings. It also gives us some insight into the basis upon which design experiences are remembered.

1 S2: Think about how heavy eggs are....

2 S4: Yeah, we need something that's going to propel this thing. I mean it's only going this far but if you think about it, it's gotta lift up 12 inches and land over there. I've got a feeling it's really gotta propel you know [motor noise] and then just go [splat noise] with a thud.

3 S1: I've got this picture in my mind of this really dramatic missile. If it's in the water, it... it could sink and it would be like a missile coming out of a submarine. [He demonstrates, pretending his pen is a missile, makes fizzing noise] ... coming out of the water, ... splashing water out.

4 S3: That reminds me cause you see those missiles come out one at... What if we did something where we sent eggs over one at a time?

5 S3: So we could have something over there to catch them like a big pillow or something I don't know, but that way you wouldn't have to launch the whole set of them. You just launch one at a time.

6 S2: Put that down: launching individually.
[S3 records idea on post-it.]
[Unrecorded conversation while flipping tape:
7 S4: We can put them each in a tennis ball.
8 S4 mentioned ping-pong ball shooters.
8 S1 didn't know what S4 was talking about.]

8 S4: Well, they're actually little springs some of them.

8 S1: Are they?

8 S4: Yeah, you know how when we were kids we could take those things that would shoot ping-pong balls and pull them back...

8 S2: I remember those! I loved those!

8 S4: ... and shoot them? Yeah. You were a deprived child.

8 S1: Were they guns?

8 S4: Yeah.

9 S4: That's actually, hmmm. That would be about the size of an egg. If we were to send it over one at a time.

10 S2: Yeah, a lot heavier, though, the eggs.

11 Later (after this meeting), S3 visualized how the idea would work and imagined that the eggs would all end up landing at the same target spot and smash each other. So S3 thought of rotating the launch mechanism so that it throws the eggs in all directions. S3 noted one interesting consequence of this was that the eggs could be thrown all at once, each in a different direction.

12 The rotating launch reminded S3 of a recently suggested idea: "flinging motion where the device is spun around and around and then let go." This had been recorded externally on a post-it.

13 This was then adapted (generalized) from having a group of eggs at the end of the string to a single egg.

14 [Two days later, this idea was discussed further while the students were going through each idea proposed so far (recorded on post-its).]

15 S3: What I was thinking was that you could just have a pole and you could have all these strings just like a May Day dance, you know where you have all the eggs hanging from strings and you spin that and the eggs all fly out and then you just let go and then they all fly.

16 S4: Now I like... that's actually pretty interesting there, cause you could .. tie them all to something like a softball...No.

17 S4: Maybe something like... I'm trying to think of something that... What about something that's squishy?

18 S4: It's gotta have... What if it has some kind of fluid, like an orange? If you put an egg inside a hollowed out orange, half hollowed out orange, each of those little things would squash, you know inside of an orange. (I just ate an orange for lunch... I bring real-life experiences to this.)

19 S1: Well, that's the concept of a shock absorber. And the way it works is... If you just have a sealed shock. If you have... What a sealed shock would be would just be a balloon. If we had the eggs sitting on top of this big balloon and it went down, whenever the balloon squashed, there'd be pressure inside the balloon and it would jump back up again, so it would bounce.
But if you have a shock absorber that has a little seal out, whenever it... it's like a balloon w/ a little tiny hole, so whenever it hits the ground, it squashes and the air shoots out so it doesn't recoil. And an orange, whenever it's squashed, the juices would go squirting out and it wouldn't rebound.

During this design episode, the students recalled many cases, most of which are devices, some in action. Two different aspects of cases seemed to get the most attention: how a device works and what are its results (i.e., what it accomplishes, how it might fail, its pros and cons). Often, what was remembered seemed to get embellished through a sort of mental simulation, sometimes causal (e.g., the operation of ping-pong ball shooter 8) and sometimes imagistic (e.g., the submarine launch 3, 4).

These remindings are used in many different ways.

1. They *generate* design ideas that can be re-used directly, adapted to the current situation, or merged with other design pieces. For example, tennis balls (7) and softballs (16) are recalled to be reused for the new purpose of protecting eggs.
2. They *predict* the outcome of proposed solutions. For example, the leaky shock absorber (19) is used to predict that an orange would not be a resilient egg protector. This is useful in *evaluating* proposed solutions.
3. They *communicate* ideas. For example, the May Day dance (15) is used to quickly communicate the structure of a design alternative.
4. They help simulate or visualize the behavior of a proposed design alternative. This is useful in *elaborating* both proposed solutions and vague, incomplete specifications. For example, S1's mental picture of a submarine submerging and launching a missile (3) is used to help simulate the desired behavior of the device being designed. Simulation and visualization are also key ways of collecting data to be used to *evaluate* a proposed solution. For example, the problem with the initial proposal to launch eggs individually, like a submarine does, was detected by mentally simulating the launch and realizing that all eggs end up at the same spot and could break each other (11).
5. Remindings can also lead to a complete *reformulation* of the problem. For example, remembering that submarines launch missiles one at a time (1) led to converting the problem from launching a group of eggs in a single launch to launching each egg individually in multiple launches.

Focus Issues

A number of focus-related issues come up as we examine the design episode above. We describe each here and discuss what seems to provide the necessary focus. In many instances, previous design cases themselves help direct the designer's attention.

Which cases are recalled?

Of all the design experiences each student designer has had, why are these particular ones recalled? In other words, on what basis are the cases recalled? For example, what made S1 recall a shock absorber (19) and use it to analyze the effectiveness of an orange as a structure to protect an egg?

A hallmark of a creative designer is variety. Given the same problem to solve several times, the creative designer might come up with several qualitatively different solutions. We hypothesize that this happens because on each occasion, the designer is reminded of different cases, knowledge, or principles for solving the problem. Each time, the designer has different cues available to use for retrieval, despite the fact that the problem itself is the same. That is, the probe to memory that recalls previous designs or design knowledge includes not only the problem specification but also aspects of the context the designer is in or has been in recently.

In the given design episode, there are a variety of types of features that form the basis for reminding. Many remindings were based on a description of the *problem*, i.e., the function or behavior desired. The submarine launching a missile (3) was recalled as an example of a device that launches from water.

The ping-pong ball shooter (8) may have been recalled by looking for a device with the desired behavior of multiple launches of individual objects. In addition to the desired behavior, prominent *visual cues* may have played a role: the rounded shape and white color of the objects to be launched could have contributed to the memory probe if S4 visualized the desired behavior.

Structural cues describing the proposed solution, or structural constraints the solution should have, often remind students of an existing device that shares those features. For example, the structure of the proposed design that flings all eggs at once on strings reminded S3 of the maypole used for May Day dances (15).

Also, *background cues* can have an effect. S4 used not only structural cues (squishy, containing fluid) to recall an orange (18), but also cues from recent or current experiences (what S4 ate for lunch). Background interests provide additional cues. S1 is planning on becoming an automotive engineer and is often reminded of designs from the automobile domain, such as the shock absorber (19).

Understanding the basis for recalling design experiences is crucial to organizing a library of design cases and choosing indices to allow access to the cases. This is discussed further in the last section.

Which features of cases are examined?

Once a relevant design case is recalled, which aspects are examined? Some lead to problem reformulations or fill in missing details of the problem specification. Some are undesirable features that suggest new constraints that should be added to the problem specification to prohibit them. Some help elaborate a proposed solution. But how is the designer's attention drawn to those aspects that can do these things?

For example, there are numerous facts associated with submarines. What drew S3's attention to the fact that they launch missiles one at a time (4), as opposed to facts about how missiles are aimed at their target or about the cramped, claustrophobic interior? Focusing on this aspect led to a complete reformulation of the problem from launching a group of eggs to launching eggs individually.

When S1 used a mental picture of a submarine launching missiles (3) to elaborate the desired behavior of the mechanism being designed, why did S1 focus on sinking and then launching, but not on other aspects of the submarines operation, such as spying on or targeting other ships using a periscope?

When S4 brought up a ping-pong shooter, first the spring mechanism responsible for shooting was considered (8). Then the weight and size of the ping-pong balls shot was considered and compared to eggs (9,10).

The reasoning goal plays a significant role in focusing attention. When S1 recalled the submarine missile launch, the team was elaborating the problem specification by describing what the mechanism should do. It was also considering the problem of launching a heavy object out of water.

In pursuing the problem elaboration goal, S1 was interested in filling in details of the behavior of the mechanism to be designed and was focused on what aspects of the submarine's launching behavior transfer over to the egg-carrying device. So S1 was drawn to coarse-grained, high-level behaviors of the submarine and missile performed when launching from water (submerging, shooting, coming out of the water). On the other hand, S3 was viewing the submarine missile launch case from the perspective of trying to borrow its solution to the launching problem. So S3's attention was drawn to the solution detail that multiple, relatively small missiles are launched one at a time. (Attention to the small nature of the missiles may have been additionally emphasized by the hand gestures S3 made in acting out the launch.)

The ping-pong ball shooter was also considered from two different viewpoints. The team considers how the gun works as part of the goal of borrowing its solution and focuses on the spring mechanism: how the spring is loaded and released. Then S4 seemed to be considering whether the gun can be reused directly. The goal of evaluating the applicability of this existing design to the current one focused S4 and S2 on the size and weight of the ping-pong balls shot, compared to eggs.

Which evaluative issues are raised?

The evaluation process checks each design option that is generated against the current design specification. It forms a critique, identifying how well the option satisfies the constraints or how badly it fails. It also notices questionable features whose desirableness or legality are unknown. In addition, a designer has goals and guidelines that are not in the initial design specification itself but whose violation or achievement can be noticed. For example, a meal planner might like meals to be easy to prepare, but may not include this in every design specification. Goel and Pirolli (1989) identify several classes of constraints that are of this nature, including domain-specific technical constraints (such as structural soundness), legislative constraints (such as building codes), common sense, pragmatic constraints (for example, "short construction time" or personal safety), and self-imposed, personal preferences (such as "not spicy").

Not all of the evaluation criteria and problem constraints are explicit at the start of the design. They gradually surface as ideas are proposed and criticized. A key focus-related issue is: of all the evaluative issues that could be raised, why do certain ones come to mind? In the ME design project, some issues were always raised. For instance, the issue of egg safety was a primary consideration, based on the initial problem statement. Others are derived from primary goals of the designers. For example, the team was to design an egg-carrying device for at least two eggs, but one student (S2) strongly advocated that the device have a high egg-carrying capacity. This meant that S2 often brought up issues concerning how well the proposed designs accommodated the weight and space required for several eggs (1, 10).

Other evaluative issues had to be discovered as ideas were proposed. One way this sometimes occurred is that features of a proposed alternative seemed to draw attention to particular issues that might not have been considered otherwise. Some of the features are more distinctive or odd and these seem to index directly into the set of implicit criteria held by the designer. For example, during the ME design project, the students were testing how well various types of spongy material cushioned eggs when dropped from two stories. A person walked by who had done a design project which also involved protecting an egg from breaking on impact. He said he wrapped the egg in a sponge soaked in motor oil and then stuffed it in a Pringles can (a narrow cardboard cylinder in which potato chips are stacked). One of the aspects that was new about this case, compared to the ideas the students had been considering is the idea of soaking the sponge in motor oil. Focusing on the motor oil aspect reminded the students of their personal preference that the device be clean. The motor oil aspect seemed to be directly associated with the cleanliness criterion.

A second way evaluative issues are discovered is through case-based projection. Previous design cases can be used to project or derive the outcome of the current one. In the design episode, S1 recognized the similarity of the orange as a cushioning "device" to a shock absorber with a leak (19) and could predict the problem of not being able to bounce back upon impact. (S1 could also explain why, based on the causal model associated with the knowledge of shock absorbers.) This helped raise the issue of resiliency (the cushioning device must be able to bounce back) upon which to criticize the orange idea (18). Navinchandra (1991) refers to this as *criteria emergence* and he models the use of cases to raise new criteria in CYCLOPS, a landscape design program.

Which problem constraints are reformulated?

During problem reformulation, how is the designer's attention drawn to particular constraints to relax or strengthen?

Turner (1991,1993) provides an initial attempt to model the problem reformulation process, which he implemented in a program called MINSTREL. Turner proposes a case-based model of creative reasoning in which a given problem is transformed into a slightly different problem and then used as a probe to a case library. A recalled solution to the new problem is then adapted back to the original problem (using solution adaptations that are associated with the problem transformations). A set of "creativity heuristics" is used to transform the problem. Examples include generalizing a constraint (and perhaps suspending it altogether), and adapting a constraint to require a related, but slightly different outcome (e.g., injuring instead of killing).

However, MINSTREL does not address important focus questions, such as what guides the problem reformulation? Which features or constraints should be adapted? We believe that incorporating feedback from the evaluation of proposed alternatives can provide focus. Evaluation can home in on what is ambiguous or vague in the problem specification and try to take advantage of new views that result from relaxing or pushing the limits of the constraints. Also, when the need to compromise arises, conflicting constraints come into focus and the designer considers how they can be changed.

In the example episode, trying to understand how a recalled design solves a pending problem (launching a heavy projectile from the water) draws attention to a constraint that can be relaxed. S3 realized that the submarine doesn't launch one heavy object, but several relatively small missiles one at a time. This revealed a constraint in the current problem (launch all eggs at once) that could be relaxed (launch each egg one at a time).

Note that the problem of focus in reformulation is not just how does a designer know which constraint of several given constraints can productively be changed. It is also one of *revealing* the constraint in the first place. The students did not think of their problem in terms of moving a *group* of eggs in a *single* launch. They assumed the eggs would be launched all at once as a group, but this assumption was not explicit. Contrasting problems solved by previous designs with the current problem is an important way to make explicit the underlying assumptions so that the designer can decide whether the assumed constraints are essential or can be lifted.

Which problem constraints are of primary importance?

Of several solutions under consideration, one might be more appropriate than the others or several might each contribute to a solution. Evaluative procedures must be able to evaluate each individual alternative by itself as well as in light of the others. Several focus questions arise: How is relative importance among the criteria decided? How are preferences among alternatives made?

Recalled cases seem to be important here. They suggest solutions, frameworks, design strategies and design philosophies, which can provide constraints with which to evaluate a solution and the preference criteria with which to prioritize the constraints. This also facilitates reformulating the specification, making trade-offs, and relaxing constraints. There may also be general and domain-specific strategies for setting priorities that we haven't discovered yet.

Priorities must be set flexibly, however. It is interesting that in the design episode, the reformulation of the original problem to one of launching eggs individually was proposed in response to the problem of launching a heavy object from water which would require a large launch force. However, the design at the end of the episode (flinging all eggs at once) lost this advantage of individual weaker launches, since it requires just as strong a launch force to launch all eggs as a group as it does to launch them individually, but in parallel. The designer must be able to opportunistically realize that a solution is good, even though it might not fit the original goals or address concerns that were primary earlier. If a positive aspect of a proposed solution makes a new constraint or goal explicit (e.g., "be entertaining" or "look neat") or solves some other pending problem, then the designer must be able to weaken the relative importance of the conflicting goals or constraints.

Summary: Lessons Learned and Open Issues

Our seven-week exploratory study broadened our understanding of the role cases can play in design. Not only are previous designs useful in generating design alternatives and in predicting the outcomes of proposed designs. They also aid evaluation, visualization, and simulation. These are key to performing the kinds of complex elaborations and reformulations of both solutions and problem specifications that are characteristic of creative design. In particular, previous design cases help address many focus issues that permeate these activities.

Understanding the role previous design cases play, the aspects that designers pay attention to, and on what basis cases are recalled helps determine a) the content of design cases and b) how to index them.

Case Content

From our observations of creative designers, we are starting to identify the types of information cases should contain. These include symbolic descriptions of a device's common functions and behaviors, its structural composition, causal descriptions of how it works, and the results of its operations, how it fails, and its pros and cons. Many of these can be encoded straightforwardly in the familiar framework of typical case descriptions, which in general capture a problem, its solution, and the outcome of the solution (Kolodner, 1993). However, there are key representational issues to be solved. One is how to encode the imagistic information that seems to be a prominent part of what is recalled and reasoned about with respect to a device. Another issue is how to capture both abstract, general knowledge about devices and more specific experiences with particular devices. The design cases must be represented on several levels of abstraction, perhaps having abstract device representations associated with several more concrete cases that represent specific experiences with the device.

Indexing

The effective use of design cases depends crucially on being reminded of the appropriate cases at the right time. By investigating the types of features that remindings are based on, we are beginning to understand how to index these design cases. Useful indices include not only the function of the associated device, its behavior, and its structure, but also prominent visual, auditory and other sensory features.

In addition, non-obvious, cross-contextual remindings (which often lead to unorthodox design alternatives) are sometimes based on abstract similarities. Other remindings are based on derived or computed features rather than available ones. An important open problem is determining which kinds of derived features tend to be most useful for design, whether there is a set of derived features that is common to design across domains, and when those features get derived.

Recent studies of creative problem solving protocols (Kolodner and Penberthy, 1990) suggest that anticipatory indexing is not sufficient to fully explain retrieval. Features that were not salient at the time a case was experienced might be important for retrieval in the current situation. Drawing new, abstract connections might be a result of re-indexing cases in terms of what is now relevant or important. We hypothesize that by continually updating the design specification, designers derive abstract connections between the current problem and similar problems (possibly in other domains). These abstractions can be used to see previous cases differently.

While working on a design problem, designers often perform sensitized recognition of current design options and objects in their environment as they reexamine and re-index ideas recently brought up or experienced. For example, in the ME design project, the students were considering using a spring launching device and went to a home improvement store to choose materials. While comparing the strengths of several springs by compressing them, they noticed that the springs bent. One student mentioned that if they were to use springs, they would have to encase the springs in collapsible tubes to prevent bending. Later, they saw a display of toilet paper holders in the store's bathroom section. They immediately recognized them as collapsible tubes which could be used to support the springs.

What is interesting is that the toilet paper holders were not immediately retrieved by the abstract index "collapsible tube." The holders had to be re-indexed under this description when they were recognized. A key to sensitized recognition is refining the description of the solution. The process of critiquing proposed ideas often yields descriptions of what an improved solution would look like: what properties it would have, what function it would provide, and what criteria it satisfies. This primes the designer to opportunistically recognize solutions in observations of the external world and in recently considered design options.

Acknowledgments. We would like to thank the members of the mechanical engineering design team, Otto Baskin, Jon Howard, and Malisa Sarntinoranont, for their invaluable cooperation. We thank Farrokh Mistree for allowing us to participate in his mechanical engineering project course and for many thought-provoking discussions throughout. We also appreciate the insightful discussions we have had with Sambasiva Bhatta, Terry Chandler, Eric Domeshek, Ashok Goel, Colin Potts, Mimi Recker, Spencer Rugaber, Terry Sargent, and Craig Zimring. This research was funded in part by NSF Grant No. IRI-8921256 and in part by ONR Grant No. N00014-92-J-1234.

References

Goel, V. and Pirolli, P. 1989. Motivating the Notion of Generic Design within Information Processing Theory: The Design Problem Space. *AI Magazine*, 10(1): 18-36.

Kolodner, J.L. 1993. *Case-Based Reasoning*. Morgan-Kaufman Publishers, Inc., San Mateo, CA.

Kolodner, J.L. and Penberthy, T.L. 1990. A Case-Based Approach to Creativity in Problem Solving. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, Cambridge, MA. August.

Navinchandra, D. 1991. *Exploration and Innovation in Design: Towards a Computational Model*. New York: Springer-Verlag.

Tong, C.H. 1988. *Knowledge-Based Circuit Design*. Ph.D. Thesis. Rutgers Technical Report LCSR-TR-108. Laboratory for Computer Science Research. Hill Center for the Mathematical Sciences Busch Campus, Rutgers University.

Turner, S. R. 1991. A Case-Based Model of Creativity. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, Chicago, 933-937.

Turner, S. R.: 1993, *MINSTREL: A Computer Model of Creativity and Storytelling*, Ph.D. Thesis. University of California at Los Angeles.

Fifteenth Annual Conference of the Cognitive Science Society, pp. 17-26, Boulder, CO, June 1993.

Creative Conceptual Change

Ashwin Ram
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
E-mail: ashwin@cc.gatech.edu

Abstract

Creative conceptual change involves (a) the construction of new concepts and of coherent belief systems, or theories, relating these concepts, and (b) the modification and extrapolation of existing concepts and theories in novel situations. I discuss these and other types of conceptual change, and present computational models of constructive and extrapolative processes in creative conceptual change. The models have been implemented as computer programs in two very different task domains, autonomous robotic navigation and fictional story understanding.

Introduction

Much research in conceptual change has focussed on developmental conceptual change in children, and scientific conceptual change in expert adults. Keil (1989), for example, is concerned with the nature of children's concepts, their differences from concepts that adults have, and how children's concepts change through cognitive development. Such conceptual change is qualitative; not only do children learn new concepts, the nature of the concepts themselves changes through development. The study of scientific conceptual change is concerned with how new conceptual structures come to replace existing conceptual structures through scientific revolutions (Kuhn, 1962) or through longer-term enterprise (Gruber, 1989). Nersessian (1991) argues that "the problem-solving strategies scientists have invented and the representational practices they have developed over the course of the history of science are very sophisticated and refined outgrowths of ordinary reasoning and representational processes."

The conceptual change that I am concerned with here is the everyday kind. It involves everyday reasoning by reasoning systems, human or machine, in situations that allow (or require) creativity and learning. Conceptual change requires two kinds of creative processes: the construction of new concepts from input information, and the extrapolation of existing concepts in novel and unfamiliar situations. The first kind of process involves reformulating low-level information, such as sensorimotor data, into higher-level abstractions. For example, a reasoner in a strange environment may improve its ability to act in that environment by learning about the effects of its actions in that environment (for example, learning to control a car on the highway). The actions themselves may be new and unfamiliar; a reasoner may need to learn about its own actions and the interactions of these actions with the environment (for example, learning to drive a car in the first place). The reasoner may also need to learn about the structure of the environment itself (for example, learning the layout of the roads in a city). All of these scenarios require creative conceptual change of a particular kind: the construction of conceptual representations to represent causal and predictive relationships between sensory inputs, motor actions, and the environment. I will call this *constructive conceptual change* since it involves the construction of new concepts from sensorimotor experience. Although this process is not usually thought of as "creative," I will argue that the process is in fact so because it results in representations that are novel, useful, and qualitatively different from those that the reasoner initially starts out with.

Another kind of process involved in creative conceptual change is that commonly associated with fictional and imaginative scenarios. Reading a science fiction story, for example, requires a temporary suspension of disbelief and the extension or adaptation of existing concepts to create a conceptual model of the described situation (which may be very different from the reasoner's real-world experience). I will call this *extrapolative conceptual change* since it involves extrapolation from existi^ng concepts to create new ones. In addition to guiding the reasoner in the current situation, the new concepts (or systems of concepts) may be useful in other contexts as well. As I will argue, the mechanisms and knowledge involved in such reasoning are not unique to understanding fiction; they are really no different from the mechanisms and knowledge involved in reasoning in nonfictional or real-world situations. Although models of creativity and conceptual change have traditionally been developed separately from models from everyday reasoning, the constructive and extrapolative processes discussed here are not viewed as being extraordinary or special; they, and the creative conceptual change that they result in, are an integral part of everyday reasoning.

Both constructive and extrapolative conceptual change have much in common with each other, as well as with developmental and scientific conceptual change. Keil (1989) argues that systematic belief systems, or "theories," are important in developmental conceptual change, and that causal relations are essential and more useful in such theories than other sorts of relations (see also Neisser, 1987). Causal belief systems are critical in extrapolative conceptual change as well since they guide and constrain the creative adaptations performed by the reasoner. Keil views concepts as partial theories in that they embody explanations or mental models of the relations between their constituents, of their origins, and of their relations to other clusters of features (see also Johnson-Laird, 1983; Murphy & Medin, 1985). Similarly, the representations constructed through extrapolative and constructive conceptual change also embody such explanations (albeit not always "correct" ones). Analogy and mental modelling play a crucial role in theories of scientific conceptual change (e.g., Nersessian, 1991), and in extrapolative conceptual change as well. All these types of conceptual change rely both on inductive and analytical reasoning processes, though sometimes to different extents. Typically, analytical processes are used when appropriate theories are available to support analysis (such as in experts), and inductive processes are used when such theories are not available (such as in novices). In addition to the creation of individual concepts and their gradual evolution through experience, conceptual change may also involve the reorganization of an entire system of concepts.

The decomposition of the processes of conceptual change into constructive and extrapolative is a functional one. Rather than discuss conceptual change in children and adults, in laypersons and scientists, or in physics and mathematics, I will focus on the underlying *functions* of conceptual change (the construction and evolution of concepts), on the *mechanisms* that achieve these functions, and on the *knowledge* that these mechanisms rely on. Such a decomposition is methodologically useful because it allows us to study the types of knowledge and processes that underlie conceptual change and their commonalities across different performance tasks, domains, and levels of expertise of the reasoners. In this paper, I will discuss computational models of constructive and extrapolative conceptual change, focussing in particular on two computer programs that instantiate the models in two very different "everyday" task domains. The computer programs aid in the development and evaluation of the models, and provide an experimental framework for further exploration of theoretical ideas. I will conclude with a discussion of a framework for the integration of these (and other) methods of conceptual change into a single "multistrategy" system.

Case studies in creative conceptual change

The computer programs presented here serve as case studies of constructive and extrapolative processes in conceptual change. The first program, called SINS (Self-Improving Navigation System) is an autonomous robotic navigation system that learns to navigate in an obstacle-ridden world (Ram & Santamaria, 1993). Autonomous robotic navigation is the task of finding a path along which a robot can physically move through a given environment and then executing the actions to carry out the movement in a real or simulated world. The ability to adapt to changes in the environment, and to learn from experiences, is crucial to adequate performance and survivability in the real world. SINS uses fast robotic control augmented with multiple learning methods that allow the system to adapt to novel environments and to learn from its experiences. The core of the system is a constructive conceptual change mechanism that autonomously and progressively constructs representational structures that encapsulate the system's experiences. These structures comprise a higher-level representation of the system's perceptual and sensorimotor interactions with its environment, and are used to aid the navigation task in two ways: they allow the system to dynamically select the appropriate robotic control behaviors in different situations, and they also allow the system to adapt selected behaviors to the immediate demands of the environment.

The second case study is based on a computer program called ISAAC (Integrated Story Analysis And Comprehension), which is a natural language understanding system that reads short stories from the science fiction genre (Moorman & Ram, 1993). Such stories require creative understanding, in which the reader must learn enough about an alien world in a short text in order to accept it as the background for the story, and simultaneously must understand the story itself. ISAAC implements a process of extrapolative conceptual change which is based on the creative extrapolation, modification, or extension of existing concepts and theories to invent new ones. The extrapolation is constrained by the content of the story, by the system's existing concepts and theories, and by the requirements of the reading and understanding task.

As the case studies will reveal, there is much in common between these two systems despite their superficial differences. Both systems use multiple types of knowledge, and multiple types of reasoning processes. Both rely on multiple sources of constraints on these processes, including theories, knowledge and knowledge organization, and actual experience. Creative conceptual change in both systems is a process of gradual evolution of concepts to create better approximations of the observed world. Both systems learn autonomously through experience. The new concepts contribute significantly to the systems' abilities to carry out their respective tasks, and may be very different from those that the systems initially started out with.

The differences between the systems are also of interest. SINS relies directly on its experiences in the real world, whereas ISAAC's real world is that of natural language texts which vicariously describe fictional world experiences of fictional characters. ISAAC integrates its processes using explicit arbitration and control; thus, conceptual change in ISAAC is guided by the particular needs and goals of the program. SINS, in contrast, learns "automatically" through its task performance, and thus is better characterized as having an implicit orientation or goal to learn (Barsalou, discussed in Leake & Ram, 1993).

The two systems are discussed in more detail below.

Constructive conceptual change

Many machine learning and conceptual change systems have traditionally been used in problem domains that can be adequately described using discrete, symbolic representations. However, an important type of conceptual change is that which occurs in continuous problem domains. In order to actually perform a task in the real world, for example, an agent (human or robot) must be able to accept perceptual or sensory inputs from the environment, select an appropriate action based on its goals, the input, and the task at hand, and then carry out that action through appropriate motor control commands to its effectors. Perception and action are inherently continuous in three ways: they require representations of continuous information, they require continuous performance (for example, driving a car), and they require continuous adaptation and learning.

For example, consider the problem of spatial representation and exploration in a real-world environment. An agent learning about its physical environment through exploration might build a cognitive map representing topological and metrical information about the space around it. Several studies have suggested that cognitive maps are organized into layers (e.g., Lynch, 1960; Piaget & Inhelder, 1967; Siegel & White, 1975). The cognitive map contains information about space, locations, connectivity, and distance, learned gradually through interaction with and exploration of the environment. These studies have motivated computational models of robot map-learning as well. For example, Kuipers & Byun (1991) describe a simulated robot, NX, that learns a hierarchy of types of spatial knowledge organized into sensorimotor, control, procedural, topological, and metrical knowledge. At the lowest level, the robot has access to raw sensory data from the environment. The robot's representation of the space surrounding it undergoes a series of conceptual changes as sensorimotor data (which is continuous and numerical) is reformulated and abstracted into successively higher-level descriptions (which are discrete and symbolic). This is an example of what I am calling constructive conceptual change in this paper.

The SINS system discussed here also learns from continuous sensorimotor information, but addresses a somewhat different problem in constructive conceptual change: that of learning the appropriate concepts for dynamic and adaptive control of action. In addition to learning about the environment around it, an agent must also learn about the interactions of its behaviors with the environment. It must learn what effects its actions have and when different actions are appropriate. This problem is different from the map-learning problem because it involves constructing representations, not just of the environment, but of the agent's interactions with the environment. Often, action and learning are incremental of necessity because the agent's knowledge is limited and because the environment is unpredictable; the agent can at best execute the most promising short-term actions available to it and then re-evaluate its progress. An agent navigating in an unfamiliar environment, for example, may not know where obstacles lie until it actually encounters them. As the problems encountered become more varied and difficult, it becomes necessary to use available knowledge in an incremental manner to act, and to rely on continuous feedback from the environment to adapt actions and learn from experiences. The problem solving and learning process must operate continuously; there is no time to "stop and think," nor a logical point in the process at which to do so. Through this on-going process, the agent must construct higher-level conceptual representations that constitute its "understanding" of the world and of its interactions with the world.

SINS addresses this problem by constructing conceptual structures that encapsulate continuous sensorimotor experience. These structures are modified continuously even as they are used to guide action. Through experience, these structures evolve into stable perception-action models and result in improved performance on a wide range of input environments.

Technical details: The SINS system

Autonomous robotic navigation is defined as the task of finding a path along which a robot can move safely from a source point to a destination point in a obstacle-ridden terrain (path planning) and executing the actions to carry out the movement in a real or simulated world (plan execution). Several methods have been proposed for this task, ranging from high-level planning methods to reactive methods.

High-level planning methods use extensive world knowledge and inferences about the environment they interact with (e.g., Fikes, Hart & Nilsson, 1972; Sacerdoti, 1975). Knowledge about available actions and their consequences is used to formulate a detailed plan before the actions are actually executed in the world. These methods can successfully perform the path-finding required by the navigation task, but only if an accurate and complete representation of the world, and of available actions and their effects, is available to the agent. Situated or reactive control methods have been proposed as an alternative to high-level planning methods (e.g., Arkin, 1989; Brooks, 1986; Kaelbling, 1986; Payton, 1986). In these methods, no planning is performed; instead, a simple sensory representation of the environment is used to select the next action that should be performed. Actions are represented as simple behaviors, which can be selected and executed rapidly, often in real-time. These methods can cope with unknown and dynamic environmental configurations, but only those that lie within the scope of predetermined behaviors.

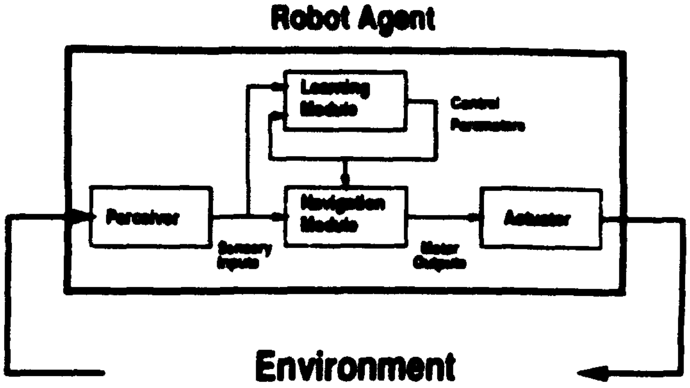


Figure 1: Architecture of the self-improving robot navigation system.

In a complex and dynamic environment, an agent needs to develop a combination of the above abilities: a fast and accurate perception process, the ability to reliably map sensory inputs to higher-level representations of the world, the ability to reliably predict the effects of its actions, and the ability to respond immediately to unexpected situations. Furthermore, to ensure adequate performance and survivability in the real world, the agent's ability to perform these functions must adapt to changes in the environment and improve through experience. In the SINS system, we have focussed on the problem of constructing representations of the agent's interactions with its environment. These representations model the environment and the effects of the agent's actions in that environment, and provide a basis for selecting appropriate actions in a possibly unfamiliar environment.

SINS uses schema-based reactive control for fast performance (Arkin, 1989), augmented with multistrategy learning methods that allow the system to adapt to novel environments and to learn from its experiences (see figure 1). The system autonomously and progressively constructs representational structures that encapsulate its experiences into "cases" that are then used to aid the navigation task in two ways: they allow the system to dynamically select the appropriate robotic control behaviors in different situations, and they also allow the system to adapt selected behaviors to the immediate demands of the environment (see figure 2).

The system's cases are automatically constructed using a hybrid case-based and reinforcement learning method without extensive high-level reasoning. The learning and navigation modules function in an integrated manner. The learning module is always trying to find a better model of the interaction of the system with its environment so that it can tune the navigation module to perform its function better. The navigation module provides feedback to the learning component so it can build a better model of this interaction. The behavior of the system is the result of an equilibrium point established by the learning module, which is trying to refine the model, and the

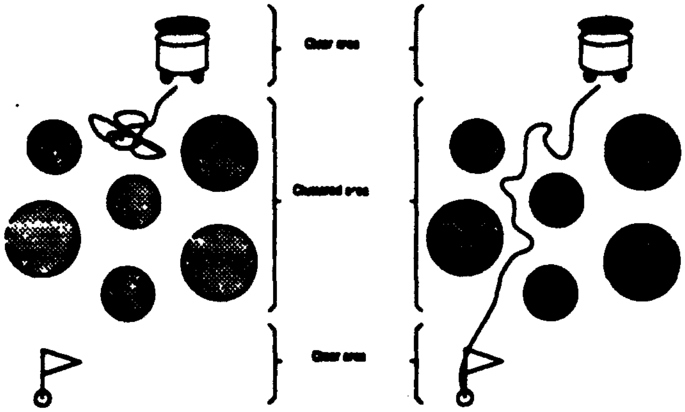


Figure 2: Typical navigational behaviors of the autonomous robotic system. The figure on the left shows the non-learning system with high obstacle avoidance and low goal attraction. On the right, the learning system has lowered obstacle avoidance and increased goal attraction, allowing it to "squeeze" through the obstacles and then take a relatively direct path to the goal.

environment. which is complex and dynamic in nature. This equilibrium may shift and need to be re-established if the environment changes drastically: however. the model is generic enough at any point to be able to deal with a very wide range of environments.

The learning methods are based on a combination of ideas from case-based reasoning and learning. which deals with the issue of using past experiences to deal with and learn from novel situations (e.g.. Hammond. 1989). and from reinforcement learning. which deals with the issue of updating the content of system's knowledge based on feedback from the environment (e.g.. Sutton. 1992). Each case in SINS represents an observed regularity between a particular environmental configuration and the effects of different actions. and prescribes the values of the control parameters that are most appropriate (as far as the system can determine based on its previous experience) for that environment.

The learning module performs the following tasks in a cyclic manner: (1) **perceive** and represent the current environment; (2) **retrieve** a case which represents an environment most similar to the current environment; (3) **adapt** the motor control parameters in use by the navigation module based on the recommendations of the case; and (4) **learn** new associations and/or adapt existing associations represented in the case to reflect any new information gained through the use of the case in the new situation to enhance the reliability of their predictions.

Since learning is not supervised by an outside expert. one of the issues to be addressed is how the system can determine whether the current experience should be used to modify and improve an existing case. or whether a new case should be created. In SINS. this is done through an inductive procedure that uses information about prior applications of the case. When a case is retrieved and applied to the current situation. a "relative similarity measure" is used to quantify how similar the current environment configuration is to the environment configuration encoded by the case. relative to how similar the environment has been in previous utilizations of the case. Intuitively. if a case matches the current situation better than previous situations it was used in. it is likely that the situation involves the very regularities that the case is beginning to capture: thus. it is worthwhile modifying the case in the direction of the current situation. Alternatively. if the match is not quite as good. the case should not be modified because that will take it away from the regularity it has been converging towards. Finally. if the current situation is a very bad fit to the case. it makes more sense to create a new case to represent what is probably a new class of situations.

A case in SINS represents a set of associations between sensory inputs and control parameters. Sensory inputs provide information about the configuration of the environment. and control parameters specify how to adapt the motor outputs of the navigation module in the environments to which the case is applicable. Each type of information is represented as a vector of analog values. Each analog value corresponds to a quantitative variable (a sensory input or a control parameter) at a specific time. and a vector of such values represents the trend or recent history of the corresponding variable. This representation has three essential properties. First. the representation is capable of capturing a wide range of possible associations between of sensory inputs and schema parameters. Second. it permits continuous progressive refinement of the associations. Finally. the representation captures trends or patterns of input and output values over time. This allows the system to detect patterns over larger time windows rather than having to make a decision based only on instantaneous values of perceptual inputs.

Sets of sensory inputs and control parameters are associated by grouping their vectors together into a single case. This grouping induces (albeit implicitly) a set of concepts that can be used to describe a control strategy or an environmental regularity. For example. if SINS is getting deeper into a crowded area. the values of the sensory inputs responsible for object detection will increase over time. A useful strategy in such a situation might be to back out and go around the obstacles. However. such a strategy cannot be expressed in purely perceptual terms; it requires the concepts of crowdedness. retreat. and so on. which are qualitatively different from the sensorimotor information that is initially available to the system.

Since learning and adaptation are based on a relative similarity measure. the overall effect of this process is to cause the cases to converge on stable associations between environment configurations and control parameters. Stable associations represent regularities in the world that have been identified by the system through its experience. and provide the predictive power necessary to navigate in future situations. The assumption behind this method is that the interaction between the system and the environment can be characterized by a finite set of causal patterns or associations between the sensory inputs and the actions performed by the system. The method allows the system to learn these causal patterns and to use them to modify its actions by updating its motor control parameters as appropriate.

One disadvantage of the analog representations is that they are not easy to interpret. making it difficult for a human observer to characterize the regularities and concepts that SINS actually learns in a given environment. To evaluate the method. we have developed a three-dimensional interactive visualization of a robot navigating through a simulated obstacle-ridden world. and used it to test the SINS system through extensive empirical simulations on a wide variety of environments using several different performance metrics. The system is very robust and can perform successfully in (and learn from) novel environments without any user intervention or supervisory input, yet it compares favorably with traditional reactive methods in terms of speed and performance (Ram & Santamaria, 1993). Furthermore, the system designers do not need to foresee and represent all the possibilities that might occur since the system develops its own "understanding" of the world and its actions.

SINS carries out a constructive conceptual change process in which new conceptual representations of regularities in system-environment sensorimotor interactions are created through experience. The process results in a qualitative shift in the system's internal "theory" of perception and action, and results in new concepts that are creative by virtue of being both original and useful (Koestler, 1964; Turner, 1991). As one might expect, the creation of new concepts in SINS (and in other systems such as NX) is an incremental process and involves, in addition to the abstraction of low-level inputs into higher-level representations, the modification of such representations in response to future experiences. In this sense, constructive conceptual change involves some degree of extrapolation as well. However, since this extrapolation does not require the kinds of creative leaps as those needed in the ISAAC system, the latter provides a better case study of extrapolative conceptual change and is discussed next.

Extrapolative conceptual change

In developing the SINS system, we were interested in the problem of constructing conceptual representations from continuous sensorimotor experience. Another type of conceptual change, however, is that which occurs when conceptual representations are used to understand a new and unfamiliar domain. The more different the domain, the more radical the change. In the ISAAC system, we are focussing on the construction of new concepts (and associated theories) through creative theory-guided transfer of existing concepts to a new domain. This process is largely analytical and involves analogical and metaphorical reasoning. There are two central issues here: what are the processes by which existing theories are extrapolated, and what is the nature of the constraints on these processes?

ISAAC explores these ideas in the domain of reading short stories from the science fiction literature. Consider the following short story, *Men Are Different* by Alan Bloch (1963).

> I'm an archaeologist, and Men are my business. Just the same, I wonder if we'll ever find out about Men—I mean *really* find out what made Man different from us Robots—by digging around on the dead planets. You see, I lived with a Man once, and I know it isn't as simple as they told us back in school.
>
> We have a few records, of course, and Robots like me are filling in some of the gaps, but I think now that we aren't really getting anywhere. We know, or at least the historians say we know, that Men came from a planet called Earth. We know, too, that they rode out bravely from star to star; and wherever they stopped, they left colonies—Men, Robots, and sometimes both—against their return. But they never came back.
>
> Those were the shining days of the world. But are we so old now? Men had a bright flame—the old word is "divine," I think—that flung them far across the night skies, and we have lost the strands of the web they wove.
>
> Our scientists tell us that Men were very much like us—and the skeleton of a Man is, to be sure, almost the same as the skeleton of a Robot, except that it's made of some calcium compound instead of titanium. Just the same, there are other differences.
>
> It was on my last field trip, to one of the inner planets, that I met the Man. He must have been the last Man in this system, and he'd forgotten how to talk—he'd been alone so long. I planned to bring him back with me. Something happened to him, though.
>
> One day, for no reason at all, he complained of the heat. I checked his temperature and decided that his thermostat circuits were shot. I had a kit of field spares with me, and he was obviously out of order, so I went to work. I pushed the needle into his neck to operate the cut-off switch, and he stopped moving, just like a Robot. But when I opened him up he wasn't the same inside. And when I put him back together I couldn't get him running again. Then he sort of weathered away—and by the time I was ready to come home, about a year later, there was nothing left of him but bones. Yes, Men are indeed different.

In order to understand this story, the reader must infer that the narrator is a robot, that robots are the dominant lifeform in the future, that humans have practically died out, that robots are capable of making logical errors such as the ones that the narrator made, and so on. The reader must construct an appropriate model of this world, and interpret the story with respect to this model even as the model evolves. The reader must also be willing to suspend disbelief to understand concepts which do not fit into a standard world view.

In ISAAC, new theories (and associated concepts) are constructed through extrapolation and modification of existing theories and concepts. The extrapolation is constrained by the actual content of the story, by the system's existing theories and concepts, and by the cognitive constraints on the reading and understanding mechanisms that are responsible for processing the story. No reader, machine or human, could have the time, memory, and other resources to read every single word in a story in-depth and to consider all the ramifications of each word. The reader's environment (the story), knowledge (existing concepts), goals and tasks (e.g., Ram & Hunter, 1992), and cognitive resources available to the processing machinery (e.g., Just & Carpenter, 1992) interact to constrain the possible extrapolation to a more manageable level.

The story understanding processes in ISAAC are not unique to science fiction stories, of course. Understanding any fictional story requires similar kinds of processing. The same is true of nonfictional stories as well as unfamiliar real-world scenarios, although the types and degree of conceptual modifications required may be different.

Technical details: The ISAAC system

The ISAAC system consists of six "supertasks," each of which is made up of several subtasks that interact with each other. The tasks are based on research in psycholinguistics (e.g., Holbrook, Eiselt & Mahesh, 1992; van Dijk & Kintsch, 1983), reading comprehension (e.g., Black & Seifert, 1981; Graesser, Golding, & Long, 1991), story understanding (e.g., Birnbaum, 1986; Ram, 1991; Rumelhart, 1977), episodic memory (e.g., Kolodner, 1984; Schank, 1982), analogy (e.g., Falkenhainer, 1987; Gentner, 1989), creativity (e.g., Gruber, 1989; Schank

& Leake, 1990), and metacognition (e.g., Gavelek & Raphael, 1985; Schneider, 1985; Weinert, 1987; Wellman, 1985). The supertasks and their functions are summarized below.

Language understanding is responsible for low-level text understanding, including lexical retrieval, syntactic parsing, pronoun reference, punctuation analysis, and tense analysis.

Story structure understanding focusses on details of the text which relate to story structure, including character identification (protagonist, antagonist), setting identification (time, location), plot description, and genre identification.

Episodic understanding carries out the event representation (agent, action, state, object, location), agent modelling (agents' goals, knowledge, and beliefs), and action modelling tasks that are central to understanding fictional, narrative or real-world episodes.

Explanation and reasoning is responsible for high-level reasoning and learning tasks, including those supporting specific language understanding tasks such as unknown word definition, and general tasks such as belief management, inference, creative analogy, interest management, and learning.

Memory management carries out memory storage and retrieval, including spontaneous reminding and case construction.

Metacontrol is responsible for integration of the other supertasks, and for focus of attention, time management, and suspension of disbelief. Since it is unreasonable to assume that the system would have complete metacognitive access to all its internal processes (Nisbett & Wilson, 1977), metacontrol and metareasoning operate on supertasks and do not access the individual tasks directly. The supertasks in turn control the individual tasks that they are responsible for.

We chose science fiction stories as the domain for ISAAC because it is a particularly good one to study what one might call "creative understanding." People can comprehend stories which have no basis in fact, and which may require invention of concepts and theories which are radically different from those in the real world. The process of understanding the un-understandable involves the extrapolative type of creative conceptual change. A central requirement is the willingness of the reader to suspend his or her disbelief of the material being presented or the assumptions being made about the fictional world (Corrigan, 1979). Consider the ambiguous title of a Larry Niven (1973) story, *Flight of the Horse*. This phrase could refer to a fleeing horse, a horse on an airplane flight, or to a flying horse. If a story understanding system relied on a belief in the validity of world knowledge, it would disambiguate the phrase to eliminate the latter meaning since it "knows" horses cannot fly. This may be incorrect if the story was about a flying horse (or a pegasus), which is perfectly reasonable in a science fiction or mythological story. As I argued earlier, these considerations are not unique to science fiction stories; even factual stories (such as newspaper stories) in domains that are not completely understood may require the system to consider the possibility that its current understanding of the domain is incomplete or incorrect (e.g., Ram, 1993).

To understand concepts which do not fit into a standard world view, the system attempts to modify existing concepts (Schank, 1986). This usually involves extending or adapting not just a single concept, but systems of concepts—that is, theories. This modification can occur in several ways. Definitional constraints may be relaxed to produce concepts with alternative constraints. For example, relaxing the definitional constraint that a horse's primary mode of locomotion is its legs may result in a "horse" with wings—a pegasus. Another option is to add new constraints or features to existing concepts, or to combine two concepts together. Suitcases, for example, do not normally have a mode of locomotion; adding one may result in an independently mobile suitcase, much like the one depicted in Terry Pratchett's (1983) story, *The Colour of Magic*. Creativity may also result from relaxed constraints on memory search processes, such as in the "imaginative memory" of Turner's (1991) MINSTREL system.

	Physical	Mental	Social	Emotional	Temporal
Agents	person	[illegible]	team	Ares	embassy
Actions	walking	thinking	selling	loving	getting closer to March
Objects	rock	idea	teacher-student relationship	hatred	second
States	young	lack of knowledge	polite distance	being angry	early

Figure 3: Knowledge representation grid.

A problem with such concept manipulation is that it is difficult to specify principled constraints on this process. Could a toaster be a good mode of horse locomotion? Up to a certain limit, constraint manipulation will result in concepts which could be called creative, after which the resulting concepts may be too bizarre to be useful. However, utility and interestingness are not inherent in particular concepts, but can only be evaluated with respect to the reasoner's knowledge, the organization of this knowledge, the reasoner's goals, the task at hand, the environment in which the reasoner is carrying out its tasks (in the case of ISAAC, the story), and general processing heuristics (Pinto, Shrager & Berthenthal, 1992; Ram, 1990).

In ISAAC, the knowledge organization scheme provides a structure for the conceptual change process. ISAAC's knowledge base is organized into a semantic network, which is indexed through a multidimensional grid (see figure 3). The rows of the grid represent "thematic roles" for adaptation; for concepts representing events, these include action, agent, state, and object. The columns of the grid represent "conceptual domains," such as physical, mental, social, emotional, and temporal. For example, a transfer is a generic action. Different types of transfers can be represented as physical (e.g., the PTRANS primitive of Schank, 1972), mental (e.g., MTRANS), and social (e.g., ATRANS). The grid also allows the system to represent emotional and temporal transfers (see also Domeshek, 1992).

Concept extrapolation is accomplished by moving around the grid, leading to creative and metaphorical interpretations of known concepts. Each type of movement incurs a cost to the system, depending on the degree to which the concept has been altered. Movement within a single cell is the easiest type to perform, movement along a single row or a single column is more difficult, and adaptations requiring movement across both rows and columns are the most difficult. Although the details of the grid are still under development, the point is that the system tries to perform the least amount of adaptation necessary, guided by the grid, such that the resulting concepts can explain and provide a structure for the input.

For example, many temporal metaphors can be represented as analogies between the physical and temporal columns of the grid (Lakoff & Johnson, 1980). In a sentence such as "Time has passed her by," for example, a temporal event is described in physical terms, and an abstract object (time) is described as the agent of the physical action. Similarly, in the second paragraph of *Men Are Different*, "we aren't really getting anywhere" is a metaphorical use of knowledge of physical actions to describe a mental action. Such a metaphor requires a larger creative leap than an adaptation within the physical column alone, such as in Schank's (1986) example in which an analogy is drawn between a jogger and a racehorse. Continuing with the earlier horse locomotion examples, a horse with wings involves an adaptation in which a known mode of locomotion (wings) is substituted for another one (legs), and is less bizarre than an independently mobile suitcase with wings in which an inanimate object is viewed as an animate agent with an invented (but plausible) mode of locomotion where none existed previously. As before, however, utility and interestingness are not absolute; a suitcase with wings (perhaps airplane wings rather than bird wings) might make sense in the right context.

In *Men Are Different*, robots, which in the real world are physical objects used as tools in manufacturing, are conceptualized as independent volitional agents. The reader must adopt this view to build an appropriate story model. Interestingly, the irony in this story derives from the fact that the robot in the story performs what one might view as the reverse inference, conceptualizing the man as a physical object to be repaired in a manner that one might use to repair a physical robotic device. It is important to note that the invented concepts are "real" within the context of the story, in contrast to the "bright flame of Men" which is metaphorical even within the fictional world. Similarly, a sentence such as "Winter is rapidly approaching" uses a spatial metaphor to describe a temporal event, whereas time travel may in fact be a "real" concept in a story. Understanding this concept involves adapting knowledge about actions, states, and causality from the physical column of the grid to the temporal. Such adaptation is the heart of the extrapolative conceptual change process. Once the new concepts and theories are built, they can be used to understand the story within the framework of these concepts; in turn, this may result in further modification of the concepts.

In addition to aiding in the story comprehension process, the new concepts and theories can also provide a basis for future problem solving in the real world (e.g., Koestler, 1964). For example, reading about a fictitious device may prompt the reader to develop a similar device in the real world, or may help the reader understand a similar device when it is actually encountered at some later point. Motorola's MicroTAC hand-held personal cellular phone, for instance, has a strong resemblance to the hand-held personal communicators used in the *Star Trek* television series. Goodman, Waterman & Alterman's (1991) SPATR system uses a similar case-based reasoning process to understand novel devices (such as an Airphone) and natural language instructions for using these devices based on hierarchical spatial models of known devices (such as an ATM). Reading about a creative problem solving episode may also allow the reader to replay the observed solution process on a real-world problem in a manner similar to Carbonell's (1986) derivational analogy.

Stories that are not creative can also be understood and used in such ways, of course. The mechanisms of conceptual change discussed here are an integral part of ordinary reasoning. Creative understanding in ISAAC is not implemented through a separate "creativity" process, but rather through normal processes of reasoning and learning (Gruber, 1989). Similarly, conceptual change in SINS also occurs through the normal processes of perception and control of action. Everyday reasoning is robust, adaptive, and creative; no special process need be postulated to model or explain these capabilities.

Discussion

On the surface, the models of constructive and extrapolative conceptual change presented above appear very different. The SINS model is inherently experiential and can be characterized as constructive induction of representations from sensorimotor input, whereas the ISAAC model is based on vicarious experience and can be characterized as theory-guided transfer of concepts to a new domain. The former is mostly inductive, whereas the latter is mostly analytical. In ISAAC, multiple processes are integrated through (some degree of) explicit arbitration and control; in SINS, the processes are automatic and the integrative control mechanisms are implicit.

It is an open question how these models which are, in some sense, at opposite ends of the spectrum of creative conceptual change might be unified into a single framework. Quine (1977) suggests that early concepts may be more perceptual, being defined inductively using an "innate similarity notion or spacing of qualities," and later concepts may become more "scientifically sophisticated," conceptual, and theory-embedded (see also Keil, 1989). Quine was interested in the issue of development of natural kinds, but perhaps a similar idea could be used to integrate perceptual and conceptual change in an "adult" reasoning system.

To facilitate integration, it is useful to look at commonalities between the models. Although the SINS model is closer to actual perceptual features in real world and the ISAAC model is closer to theories and mental models, both are based in real experience (whether personal or vicarious), and are constrained by the interaction between the system and the environment. Both are creative processes, and result not just in learning but in conceptual change as well. In SINS, raw sensorimotor information is encapsulated into predictive perception-action models, and in ISAAC, existing theories are modified to provide a belief structure for new and unfamiliar concepts. Both require inductive and analytical processes (although to different degrees), and both combine multiple methods of learning, concept formation, and conceptual change. Both are based on multiple types of knowledge. In both, existing knowledge provides constraints on reasoning and learning processes. Both types of creative conceptual change model a gradual evolution of concepts to better approximate the observed world and in both, evolving concepts are used in the performance task even as they are modified. These points also highlight many of similarities between the models of constructive and extrapolative conceptual change presented here and other models of conceptual change, including models of developmental and scientific conceptual change.

One framework for integration of these (and other) methods of conceptual change is through a multistrategy learning model, in which various learning methods are combined into a unified framework. Recent attention to such models is evident in machine learning (e.g., Carbonell, Knoblock & Minton, 1991; Michalski & Tecuci, 1993) and cognitive psychology (e.g., Anderson, 1983; Wisniewski & Medin, 1991). Multistrategy approaches provide the flexibility and power required in practical, real-world domains.

There are several methods of integrating multiple learning algorithms into a single system (see Michalski & Tecuci, 1993). One such framework is that used in the Meta-AQUA and Meta-TS systems (Ram & Cox, 1993; Ram, Cox & Narayanan, 1992). In this model, the reasoning system actively selects and combines learning methods based on an analysis of its learning goals which are represented explicitly in the system. Some learning goals may be low-level and always active, such as in SINS and NX. These systems can be described as performing "goal relevant" learning, in that learning is relevant to the overall goals of the system (Thagard) but the system only has an implicit goal to learn (Barsalou, both discussed in Leake & Ram, 1993). Other learning goals may be selected based on a higher-level analysis of utility of knowledge and relevance to the system's tasks, such as in ISAAC, Meta-AQUA, IVY (Hunter, 1990), and PAGODA (desJardins, 1992). These systems are better described as "goal directed" since goals are explicitly represented and used to drive the selection and execution of reasoning and learning strategies (Leake & Ram, 1993; Ram & Cox, 1993; Ram & Hunter, 1992).

Although I do not want to suggest that humans have perfect or conscious metacognitive knowledge of, and control over, their learning processes, such a model could be used to take an intentional stance (Dennett, 1987) towards a computational theory of multistrategy reasoning, both as a description of human reasoning processes and as a basis for the design of creative AI systems. Meta-TS, for example, implements a computational model of human operators learning to troubleshoot physical devices (Narayanan & Ram, 1992). The model is based on observations of human troubleshooting operators and protocol analysis of the data gathered in the test area of an operational electronics assembly manufacturing plant. In Meta-TS, multiple learning methods for knowledge compilation (Anderson, 1989), interactive transfer of expertise (Davis, 1979), postponement (Ram, 1991, 1993), and forgetting are integrated through metacognitive analysis. Experimental results in metacognition also suggest that such analysis can facilitate reasoning and learning (e.g., Alexander, 1992; Carr, 1992; Schneider, 1985; Weinert, 1987). An open issue in the design of such models is the integration of "automatic" learning strategies, such as those used in SINS, that are goal-relevant rather than explicitly goal-directed.

In conclusion, creative conceptual change is an everyday process involving multiple integrated mechanisms that are constrained by existing knowledge and by the task at hand. This process is situated in, and therefore also constrained by, the real world, and results in original, useful and qualitatively different representations of systems of beliefs. The process involves the on-going construction and extrapolation of concepts and theories in the context of, in service of, and in response to a real-world performance task. The constructive and extrapolative processes are modelled computationally through specification of functions (tasks), mechanisms and knowledge; these models are then instantiated as computer programs and evaluated empirically. In this paper, I have used robotic navigation in dynamic environments and comprehension of actual science fiction short stories as the task domains in which to present two case studies of creative conceptual change. These case studies highlight the issues involved in conceptual change, provide a basis for the development and evaluation of models that address these issues, and raise several new issues for future research.

Acknowledgements

Juan Carlos Santamaria and Kenneth Moorman contributed to the research and program implementations described here. I would also like to thank Mike Cox and Eric Domeshek for their comments on an earlier draft of this paper.

References

Alexander, J. (1992). Metacognition and Giftedness. Paper presented at *The SouthEast Cognitive Science Conference*, abstracted in Technical Report #1, Cognitive Science Program, Georgia Institute of Technology, Atlanta, GA.

Anderson, J.R. (1983). *The Architecture of Cognition*. Harvard University Press, Cambridge, MA.

Anderson, J.R. (1989). A Theory of the Origins of Human Knowledge. *Artificial Intelligence*, 30:313–351.

Arkin, R.C. (1989). Motor Schema-Based Mobile Robot Navigation. *The International Journal of Robotics Research*, 8(4):92–112.

Birnbaum, L. (1986). *Integrated Processing in Planning and Understanding*. Ph.D. thesis, Research Report #489, Yale University, Department of Computer Science, New Haven, CT.

Black, J.B. & Seifert, C.M. (1981). The Psychological Study of Story Understanding. Technical Report #18, Yale University, New Haven, CT.

Brooks, R. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23.

Bloch, A. (1963). Men Are Different. In I. Asimov & G. Conklin, editors, *50 Short Science Fiction Tales*, Macmillan Publishing Company, New York.

Carbonell, J.G. (1986). Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. In R.S. Michalski, J.G. Carbonell, & T.M. Mitchell, editors, *Machine Learning II: An Artificial Intelligence Approach*. Morgam Kaufman Publishers, San Mateo, CA.

Carbonell, J.G., Knoblock, C.A., & Minton, S. (1991). PRODIGY: An Integrated Architecture for Planning and Learning. In K. Van Lehn, editor, *Architectures for Intelligence*, pages 241–278, Lawrence Erlbaum Associates, Hillsdale, NJ.

Carr, M. (1992). Metacognitive Knowledge as a Predictor of Decomposition Strategy Use. Paper presented at *The SouthEast Cognitive Science Conference*, abstracted in Technical Report #1, Cognitive Science Program, Georgia Institute of Technology, Atlanta, GA.

Corrigan, R.W. (1979). *The World of the Theatre*. Scott, Foresman and Company, Glenview, IL.

Davis, R. (1979). Interactive Transfer of Expertise: Acquisition of New Inference Rules. *Artificial Intelligence*, 12:121–157.

Dennett, D. (1987). *The Intentional Stance*. Bradford Books/ MIT Press, Boston, MA.

desJardins, M. (1992). Goal-Directed Learning: A Decision-Theoretic Model for Deciding What to Learn Next. In *Proceedings of the ML-92 Workshop on Machine Discovery*, pages 147–151, Ninth International Machine Learning Conference, University of Aberdeen, Scotland.

Domeshek, E.A. (1992). *Do the Right Thing: A Component Theory for Indexing Stories as Social Advice*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT.

Falkenhainer, B. (1987). Scientific Theory Formation through Analogical Inference. In *Proceedings of the Fourth Inter-*

national Workshop on Machine Learning. Morgan Kaufman Publishers. Los Altos. CA.

Fikes. R.E.. Hart. P.E.. & Nilsson. N.J. (1972). Learning and Executing Generalized Robot Plans. *Artificial Intelligence*. 3:251–288.

Gavelek. J.R. & Raphael. R.E. (1985). Metacognition. Instruction and the Role of Questioning Activities. In D.L. Forrest-Pressley. G.E. MacKinnon. & T.G. Waller. editors. *Metacognition. Cognition. and Human Performance. Volume 2 (Instructional Practices)*. pages 103–136. Academic Press. New York.

Gentner. D. (1989). Mechanisms of Analogical Learning. In S. Vosniadou & A. Ortony. editors. *Similarity and Analogical Reasoning*. Cambridge University Press. London.

Goodman. M.. Waterman. S.. & Alterman. R. (1991). Interactive Reasoning about Spatial Concepts. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*. pages 734–738. Lawrence Erlbaum Associates. Hillsdale. NJ.

Graesser. A.. Golding. J.M.. & Long. D.L. (1991). Narrative Representation and Comprehension. In R. Barr. M.L. Kamil. P. Mosenthal & P.D. Pearson. editors. *Handbook of Reading Research*. volume 2. chapter 8. Longman Publishing Group. White Plains. NY.

Gruber. H.E. (1989). The Evolving Systems Approach to Creative Work. In D.B. Wallace & H.E. Gruber. editors. *Creative People at Work*. pages 3–24. Oxford University Press. New York.

Hammond. K.J. (1989). *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press. Boston. MA.

Holbrook. J.K.. Eiselt. K.P.. & Mahesh. K. (1992). A Unified Process Model of Syntactic and Semantic Error Recovery in Sentence Understanding. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*. pages 195–200. Lawrence Erlbaum Publishers. Hillsdale. NJ.

Hunter. L.E. (1990). Planning to Learn. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*. pages 261–276. Boston. MA.

Johnson-Laird. P.N. (1983). *Mental Models: Towards a Cognitive Science of Language. Inference. and Consciousness*. Harvard University Press. Cambridge. MA.

Just. M.A. & Carpenter. P.A. (1992). A Capacity Theory of Comprehension: Individual Differences in Working Memory. *Psychological Review*. 99(1):122–149.

Kaelbling. L. (1986). *An Architecture for Intelligent Reactive Systems*. Technical Note #400. SRI International.

Keil. F.C. (1989). *Concepts, Kinds, and Cognitive Development*. MIT Press, Cambridge, MA.

Koestler. A. (1964). *The Act of Creation*. MacMillan Publishers. New York.

Kolodner. J.L. (1984). *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model*. Lawrence Erlbaum Associates. Hillsdale. NJ.

Kuhn. T.S. (1962). The Structure of Scientific Revolutions. University of Chicago Press. Chicago. IL.

Kuipers. B.J. & Byun. Y.-T. (1991). A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations. *Robotics and Autonomous Systems*. 8(1-2):47–63.

Lakoff. G. & Johnson. M. (1980). *Metaphors We Live By*. University of Chicago Press. Chicago. IL.

Leake. D. & Ram. A. (1993). *Goal-Driven Learning: Fundamental Issues and Symposium Report*. Technical Report #85. Indiana University. Cognitive Science Program. Bloomington. IN.

Lynch. K. (1960). *The Image of the City*. MIT Press. Cambridge. MA.

Michalski. R.S. & Tecuci. G. (1993). editors. *Machine Learning: A Multistrategy Approach. Volume IV*. Morgan Kaufman Publishers. San Mateo. CA. (In press.)

Moorman. K. & Ram. A. (1993). A New Perspective on Story Understanding. In *Proceedings of the Thirty-First Southeast ACM Conference*. Birmingham. AL.

Murphy. G.L. & Medin. D.L. (1985). The Role of Theories in Conceptual Coherence. *Psychological Review*. 92:289–316.

Narayanan. S. & Ram. A. (1992). Learning to Troubleshoot in Electronics Assembly Manufacturing. In *Proceedings of the ML-92 Workshop on Integrated Learning in Real-World Domains*. Ninth International Machine Learning Conference. University of Aberdeen. Scotland.

Neisser. U. (1987). editor. *Concepts and Conceptual Development: Ecological and Intellectual Factors in Categorization*. Cambridge University Press. 1987.

Nersessian. N.J. (1991). How Do Scientists Think? Capturing the Dynamics of Conceptual Change in Science. In R.N. Giere. editor. *Minnesota Studies in the Philosophy of Science. Volume XV (Cognitive Models of Science)*. University of Minnesota Press. Minneapolis. MN.

Nisbett. R.E. & Wilson. T.D. (1977). Telling More Than We Can Know: Verbal Reports on Mental Processes. *Psychological Review*. 84(3).

Niven. L. (1973). *The Flight of the Horse*. Ballantine Books. New York.

Payton. D. (1986). An Architecture for Reflexive Autonomous Vehicle Control. In *Proceedings of the 1986 IEEE Conference on Robotics and Automation*. pages 1838–1845. IEEE.

Piaget. J. & Inhelder. B. (1967). *The Child's Conception of Space*. Norton Publishers. New York.

Pinto. J.. Shrager. J. & Berthenthal. B.I. (1992). Developmental Changes in Infants' Perceptual Processing of Biomechanical Motions. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*. pages 60–65. Lawrence Erlbaum Associates. Hillsdale. NJ.

Pratchett. T. (1983). *The Colour of Magic*. Penguin Books. New York.

Quine. W.V.O. (1977). Natural Kinds. In S.P. Schwarts. editor. *Naming, Necessity, and Natural Kinds*. Cornell University Press. Ithaca. NY.

Ram. A. (1990). Knowledge Goals: A Theory of Interestingness. In *Proceedings of the Twelvth Annual Conference of the Cognitive Science Society*, pages 206–214, Lawrence Erlbaum Associates. Hillsdale. NJ.

Ram. A. (1991). A Theory of Questions and Question Asking. *The Journal of the Learning Sciences*. 1(3&4):273–318.

Ram. A. (1993). Indexing. Elaboration and Refinement: Incremental Learning of Explanatory Cases. *Machine Learning*. 10:201-248. (In press.)

Ram. A. & Cox. M.T. (1993). Introspective Reasoning using Meta-Explanations for Multistrategy Learning. In R.S. Michalski & G. Tecuci. editors. *Machine Learning: A Multistrategy Approach. Volume IV*. Morgan Kaufman Publishers. San Mateo. CA. (In press.)

Ram. A.. Cox. M.T.. & Narayanan. S. (1992). An Architecture for Integrated Introspective Learning. In *Proceedings of the ML-92 Workshop on Computational Architectures for Supporting*

Machine Learning and Knowledge Acquisition, Ninth International Machine Learning Conference, University of Aberdeen, Scotland.

Ram, A. & Hunter, L. (1992). The Use of Explicit Goals for Knowledge to Guide Inference and Learning. *Applied Intelligence*, 2(1):47–73.

Ram, A. & Santamaria, J.C. (1993). A Multistrategy Case-Based and Reinforcement Learning Approach to Self-Improving Reactive Control Systems for Autonomous Robotic Navigation. In *Proceedings of the Second International Workshop on Multistrategy Learning*. Center for Artificial Intelligence, George Mason University, Fairfax, VA. (To appear.)

Rumelhart, D.E. (1977). Understanding and Summarizing Brief Stories. In D.L. Berge and J. Samuels, editors, *Basic Processes in Reading and Comprehension*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Sacerdoti, E.D. (1975). *A Structure for Plans and Behavior*. Technical Note #109, SRI International. Summarized in P.R. Cohen & E.A. Feigenbaum, *Handbook of AI, Volume III*, pages 541–550.

Schank, R.C. (1972). Conceptual Dependency: A Theory of Natural Language Understanding. *Cognitive Psychology*, 3(4):552–631.

Schank, R.C. (1982). *Dynamic Memory: A Theory of Learning in Computers and People*. Cambridge University Press, New York.

Schank, R.C. (1986). *Explanation Patterns: Understanding Mechanically and Creatively*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Schank, R.C. & Leake, D.B. (1990). Creativity and Learning in a Case-Based Explainer. In J.G. Carbonell, editor, *Machine Learning: Paradigms and Methods*. MIT Press, Cambridge, MA.

Schneider, W. (1985). Developmental Trends in the Metamemory-Memory Behavior Relationship: An Integrative Review. In D.L. Forrest-Pressley, G.E. MacKinnon, & T.G. Waller, editors, *Metacognition, Cognition, and Human Performance, Volume 1*. Academic Press, New York.

Siegel, A.W. & White, S.H. (1975). The Development of Spatial Representations of Large-Scale Environments. In H.W. Reese, editor, *Advances in Child Development and Behavior*. Academic Press, New York.

Sutton, R.S. (1992), editor. *Machine Learning*, 8(3/4), special issue on Reinforcement Learning.

Turner, S.R. (1991). A Case-Based Model of Creativity. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 933-937, Lawrence Erlbaum Associates, Hillsdale, NJ.

van Dijk, T.A. & Kintsch, W. (1983). *Strategies of Discourse Comprehension*. Academic Press, New York.

Weinert, F.E. (1987). Introduction and Overview: Metacognition and Motivation as Determinants of Effective Learning and Understanding. In F.E. Weinert & R.H. Kluwe, editors, *Metacognition, Motivation, and Understanding*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Wellman, H.M. (1985). The Origins of Metacognition. In D.L. Forrest-Pressley, G.E. MacKinnon, & T.G. Waller, editors, *Metacognition, Cognition, and Human Performance, Volume 1*, Academic Press, New York.

Wisniewski, E.J. & Medin, D.L. (1991). Harpoons and Long Sticks: The Interaction of Theory and Similarity in Rule Induction. In D. Fisher & M.J. Pazzani, editors, *Concept Formation: Knowledge and Experience in Unsupervised Learning*. Morgan Kaufman Publishers, San Mateo, CA.

To appear in a Special issue of the International Journal AI EDAM on "Machine Learning in Design", 1993

Discovery of Physical Principles from Design Experiences*

Sambasiva R. Bhatta and **Ashok K. Goel**
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
bhatta@cc.gatech.edu, goel@cc.gatech.edu

Abstract

One method for making analogies is to access and instantiate abstract domain principles, and one method for acquiring knowledge of abstract principles is to discover them from experience. We view generalization over experiences in the absence of any prior knowledge of the target principle as the task of hypothesis formation, a subtask of discovery. Also, we view the use of the hypothesized principles for analogical design as the task of hypothesis testing. another subtask of discovery. In this paper, we focus on discovery of physical principles by generalization over design experiences in the domain of physical devices. Some important issues in generalization from experiences are what to generalize from an experience, how far to generalize, and what methods to use. We represent a reasoner's comprehension of specific designs in the form of structure-behavior-function (SBF) models. An SBF model provides a functional and causal explanation of the working of a device. We represent domain principles as device-independent behavior-function (BF) models. We show that (i) the function of a device determines what to generalize from its SBF model, (ii) the SBF model itself suggests how far to generalize, and (iii) the typology of functions indicates what method to use.

1 Introduction

Analogy, as it is commonly accepted, plays an important role in reasoning. Making analogies, however, is not always easy due to the difficulty of retrieving the right analog from memory and deciding on what to transfer from the retrieved analog to the problem at hand. One

*An earlier, shorter paper appeared in the Proceedings of AID'92 workshop on "Machine Learning in Design," June 1992, Pittsburgh, USA, and in the Proceedings of ML'92 workshop on "Machine Discovery," July 1992, Aberdeen, Scotland.

method of analogical transfer is to directly map the analog to the current problem [Gentner, 1983]. This method also forms the basis of much recent work in case-based reasoning [Kolodner and Simpson, 1989; Riesbeck and Schank, 1989; Hammond, 1989; Rissland and Ashley, 1987; Alterman, 1988]. For example, in our earlier work on case-based design, we showed how *structure-behavior-function* (SBF) models of physical devices can be used for directly mapping the designs of those devices to new problems [Goel, 1991a]. An SBF model of a device captures the reasoner's comprehension of how the device works, that is, how the structure of its design results in its output behaviors. While such methods can be very useful for making analogies within a given domain, cross-domain transfer often requires higher-level abstractions such as domain principles. Since physical principles and processes typically do not refer to any specific device, we represent them as *behavior-function* (BF) models. In this model-based method for analogical transfer, new problems are solved by accessing and instantiating the BF models of principles and processes.

An important issue in model-based analogy is how to acquire knowledge of domain principles. One solution is to acquire them from a teacher, which, in fact, is a common method for acquiring such knowledge. Another method is to incrementally *discover* the principles from experiences. For example, auto mechanics apparently learn principles of automobile engineering from their experiences with auto repair although they do not always start with a deep understanding of the domain. Similarly, electronic hobbyists often learn about electrical processes from their experiences in designing electronic circuits and designers of heating and cooling equipment might acquire an understanding (i.e., a model) of how "heat exchange" occurs and what is "heat flow." By discovery, we mean learning a "concept" description from examples without knowing the target concept *a priori*. This is unlike most explanation-based learning systems [DeJong and Mooney, 1986; Mitchell *et al.*, 1986] that assume some knowledge of the target concept that needs to be learned.

The process of discovery is generally considered to have two distinct phases [Klahr and Dunbar, 1988]: *hypothesis formation* and *hypothesis testing*. One method for hypothesis formation is to incrementally generalize over design experiences. The use of a generalization in analogy acts as a test for the hypothesis. Depending on the feedback from this evaluation, the hypothesis may get revised (generalized further or refined).

In this paper, we focus on the formation of hypotheses about physical principles such as the "zeroth law of thermodynamics" from experiences in designing physical devices and briefly touch upon how they can be used for analogical design. This law states that when a hot body is brought in thermal contact with a cold body, heat flows from the hot body to the cold body [Fermi, 1937]. We show how the BF models of physical principles can be acquired by a gradual removal of structural information from the SBF models of specific devices. This process of generalization occurs while storing a design case for potential reuse. Kerr and Duffy [1992] consider generalization of past designs as one way of rationalizing past design knowledge such that it is useful in later design.

Generalization from experiences raises three important issues:

1. **The issue of relevance:** This is the issue of deciding what to generalize from an experience. With respect to this issue, the method of pure induction over design experiences could potentially become complex. Hence there is a need for developing more efficient and effective learning methods that can bias the learning in design and reduce its complexity. We show that the specification of the function of a new device can help determine what to generalize from its SBF model, and thus alleviate the problem of complexity with subsequent induction.[1]

2. **The issue of level of generalization:** This is the issue of determining how far to generalize a chosen aspect of the device. We show that the SBF model together with the knowledge of design objects, such as components and substances, can help determine how far to generalize.

3. **The issue of method selection:** This is the issue of deciding what methods to use for generalization. We show that a typology of device functions can help to determine what strategy to use.

Figure 1 presents the learning task we analyzed in this paper.

[1] In applying induction, most existing methods in machine learning assume that the instances/examples for induction are available in batch; in contrast. our model-based method relaxes this assumption and allows for experiences to come in incrementally.

Input:
- Design Experience [consisting of design problem (i.e.. function), solution (i.e., structure), and explanation (i.e., SBF model)].
  e.g., design of a sulfuric acid cooler.

Output:
- Generic principles of the domain (represented as BF models).
  e.g., the zeroth law of thermodynamics.

Method:
- Model-based generalization with inductive biasing.
  e.g., function of a design determines what parts of the experience to focus on.

Knowledge:
- Typology of primitive functions in the domain.
  e.g., ALLOW, PUMP.
- Typology of functions in the domain (consisting of primitive functions).
  e.g., substance-parameter transformation.
- Substances in the domain.
  e.g., nitric acid, water.
- Components in the domain.
  e.g., pipe, chamber.
- Past design experiences in memory.
  e.g., design of a nitric acid cooler.

Figure 1: **Learning task analyzed in this paper**

The proposed model-based method(s) for discovering physical principles from design experiences is implemented as a learning component of IDEAL,[2] an integrated system for design by analogy and learning of abstract models, that designs physical devices such as electrical circuits and heat exchangers.

2 Design Experience

IDEAL takes as input a specification of a function of the desired design and gives as output a structure that realizes the specified function and an SBF model that explains how the structure realizes that function. A design case in IDEAL specifies (i) the functions delivered by the stored design, (ii) the structure of the design, and (iii) a pointer to the causal behaviors of the design (SBF model). Since IDEAL solves *function-to-structure* design tasks, cases are indexed by the functions that the stored designs can deliver. The design cases are organized along multiple dimensions of generalization where the dimensions pertain to the constituents

[2] IDEAL stands for Integrated "DEsign by Analogy and Learning."

of design functions.

The problem-solving component of IDEAL evolves from KRITIK, an integrated case-based and model-based design system [Goel, 1991a; Goel, 1992]. Given the functional specification of a desired design, IDEAL retrieves the closest matching case from the case memory. Then it uses the SBF model of the selected design to adapt the design structure so as to meet the given functional specification. Next it revises the SBF model of the old design to incorporate the structural modifications and generates an SBF model for the new design. IDEAL uses repair plans for modifying a selected design. It verifies the new design by a qualitative simulation of the new SBF model. Finally, the new design case generated by IDEAL acts as input to its learning component. IDEAL first learns indices to the new design case [Bhatta and Goel, 1992] and stores the design for potential reuse. While storing design cases it notices similarities between the SBF models of specific designs in memory and discovers principles as described later in this article. The learned principles are intended to be evaluated and revised by their use in cross-domain analogical design.

3 Device Models

IDEAL's functional models of specific devices are represented in the form of structure-behavior-function (SBF) models. These models are based on a *component-substance ontology* [Bylander and Chandrasekaran, 1985]. This ontology gives rise to a representation language [Goel, 1992] for describing the SBF model of a design that is a generalization on Sembugamoorthy and Chandrasekaran's [1986] functional representation scheme. The constituents of the SBF model are described below.

Structure: The structure of a design is expressed in terms of its constituent components and substances and the interactions between them. Figure 2 shows the structure of a `sulfuric acid cooler` (SAC) schematically.

Function: A function is represented as a schema that specifies the behavioral state the function takes as input, the behavioral state it gives as output, and a pointer to the internal causal behavior of the design that achieves the function. Figure 3(a) shows a function of the `SAC`, namely, heating water. The input state of the function specifies that water at `location`

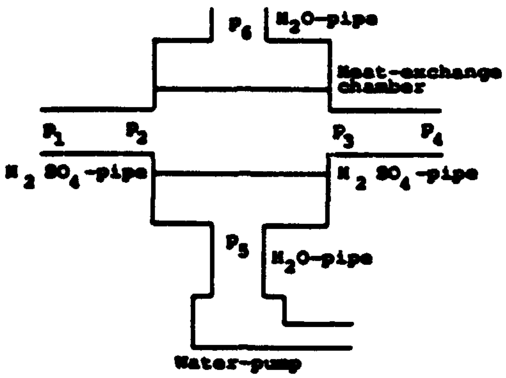
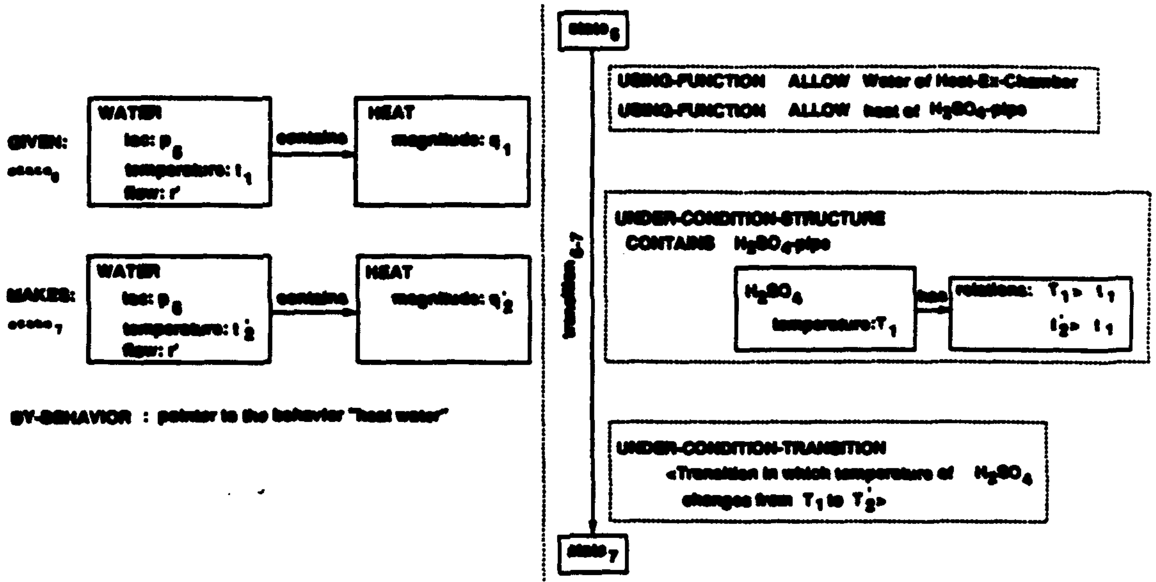


Figure 2: **Sulfuric Acid Cooler**

p5 in the topography of the device (Figure 2) has the properties **temperature** and **flow**, and corresponding parameters t_1 and r'. It also specifies that the water contains another substance **heat** whose **magnitude** is q_1. Similarly, the output state specifies the properties and the corresponding parameters of the substance at **location** p6.

This representation of functions gives rise to a typology of functions in the domain: transformation functions, control functions, maintenance functions, etc. In this article, we will be focusing on transformation functions, which again are of several types: substance transformation, substance-parameter transformation, and substance-location transformation. For example, the function of SAC is both a *substance-parameter transformation* and a *substance-location transformation* because it specifies a change in the parameter of the substance temperature as well as a change in the substance location.

Behavior: The internal causal behaviors of a device are viewed as sequences of *state transitions* between *behavioral states*. The annotations on the state transitions express the *causal, structural,* and *functional context* in which the transformation of state variables, such as substance, location, properties, and parameters, can occur. The causal context provides *causal relations* between the variables in preceding and succeeding states. The structural context specifies different kinds of structural information such as substances, components, structural relations among components and substances, and spatial locations in the device. The functional context indicates which functions of components in the device are responsible for the transition. Figure 3(b) shows the causal behavior that explains how water is heated


GIVEN:
state$_6$
WATER
loc: p$_5$
temperature: t$_1$
flow: r'
contains
HEAT
magnitude: q$_1$
MAKES:
state$_7$
WATER
loc: p$_6$
temperature: t$_2$'
flow: r'
contains
HEAT
magnitude: q$_2$'
BY-BEHAVIOR : pointer to the behavior "heat water"
state$_6$
transition$_{6-7}$
USING-FUNCTION ALLOW Water of Heat-Ex-Chamber
USING-FUNCTION ALLOW heat of H$_2$SO$_4$-pipe
UNDER-CONDITION-STRUCTURE
CONTAINS H$_2$SO$_4$-pipe
H$_2$SO$_4$
temperature:T$_1$
has
relations: T$_1$ > t$_1$
t$_2$' > t$_1$
UNDER-CONDITION-TRANSITION
<Transition in which temperature of H$_2$SO$_4$
changes from T$_1$ to T$_2$'>
state$_7$

(a) Function "Heat Water" of SAC

(b) Behavior "Heat Water" of SAC

Note: All locations are with reference to components in this design.
All labels for states and transitions are local to this design.

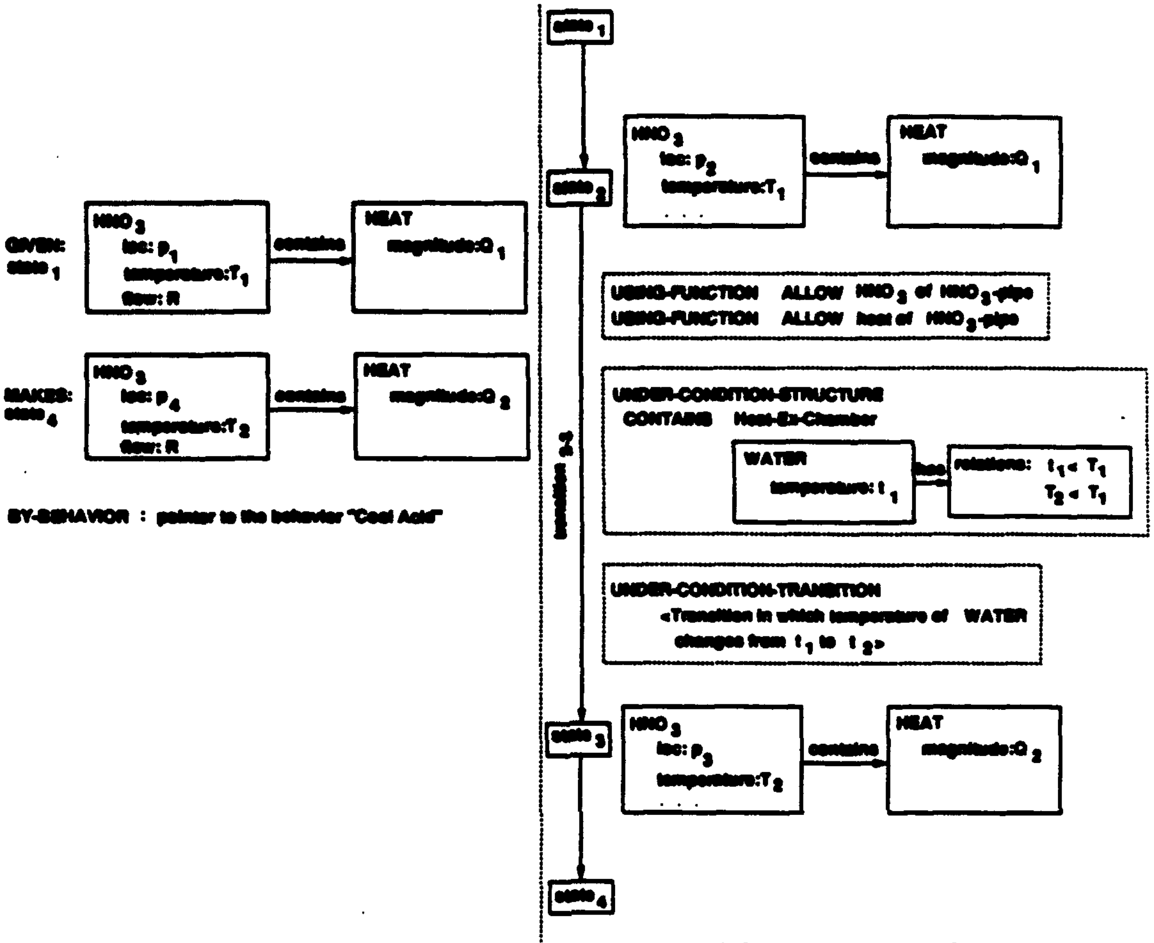
Figure 3: Function and Behavior of Sulfuric Acid Cooler

from **temperature** t_1 to t_2'. *State*$_6$, the preceding state of *transition*$_{6-7}$, describes the state of water at location p5 and *state*$_7$, the succeeding state, at location p6.

The UNDER-CONDITION-STRUCTURE annotation in *transition*$_{6-7}$ specifies that the behavior **allow** of H_2SO_4-pipe can allow the flow of heat only if the H_2SO_4-pipe CONTAINS sulfuric acid with a temperature of T_1 that is greater than t_1. The qualitative parameter relations on the substance properties. such as those shown for **temperature** in Figure 3(b). are a crucial part of describing the causal process underlying a transition.

4 A Model-Based Method for Hypothesis Formation

Consider, for example, the situation in which IDEAL finds multiple (e.g., two) cases to be similar in their functions while it is storing a design case in the functionally organized case memory. We will consider the designs of sulfuric acid cooler and nitric acid cooler whose function and behavior are shown respectively in Figures 3 & 4 for the purpose of illustrating the methods. The similarity between two functions is determined by comparing the input state and output state in them. Furthermore, similarity between two states is determined



(a) Function "Cool Acid" of NAC

(b) Behavior "Cool Acid" of NAC

Note: All locations are with reference to components in this design.
All labels for states and transitions are local to this design.

Figure 4: **Function and Behavior of Nitric Acid Cooler**

by comparing different slots in the schemas, such as **substance**, **location**, and other properties. For instance, a function F_1 is more similar to another function F_2 than it is to F_3 if the **substance** in both F_1 and F_2 is same while it is different in F_3. For example, the function of a nitric acid cooler that cools nitric acid from T_1 to T_2 is more similar to another nitric acid cooler that cools nitric acid from T_1 to T_3 than it is to a sulfuric acid cooler that cools sulfuric acid from T_1 to T_2. This is based on the heuristic that changing a substance altogether in a design is harder than changing a property of a substance. These similarity measures are based on those used in KRITIK for accessing cases from memory [Goel, 1992]. In addition to generalizing the functions of similar design cases, IDEAL can also generalize the associated SBF models for use in solving problems by analogy in a different domain with the experience gained in one domain. However, IDEAL does not know *a priori* what the target "concept" will be; hence, it formulates the generalized model as a hypothesis.

As mentioned earlier, the function of a device determines what parts of its model to generalize. If the function is a *transformation function* (e.g., substance transformation, substance-parameter transformation, substance-location transformation) then any relations in the different types of context annotating the transitions in the behavior that describe the corresponding change and the transitions themselves can be generalized to form meaningful abstractions of behaviors. For example. since the function "heat water" of sulfuric acid cooler is to transform the temperature of the substance water from one value to another, the transition *transition*$_{6-7}$ in Figure 3(b) is useful to focus on. The relations on the parameters of temperature describing the change can be generalized along with the similar behavior of another cooler or heater. In addition to the parametric relations, other aspects of the context, such as conditions on substance and conditions on structural relations that involve the parameter being transformed, also form an important part of the content to be generalized.

After identifying what parts of the specific models to focus on, the issue is to determine what kinds of changes along a dimension are meaningful for generalization. In other words, the issue is what similarities between the two models (in the focused segments of the behaviors) are retained, as they are, in the generalization and what differences are generalized. The same kind of similarity metrics as those for comparing functions are used for this purpose as well, because a focused segment of behavior includes a sequence of states and state-transitions. However, in addition to comparing states, the annotations on the transitions are also compared as guided by the functions (explained above). Since generalizations tend to deal with more qualitative parameters than specializations, we consider *positive* changes (i.e., increase) and *negative* changes (i.e., decrease) in the parameter of the chosen property for generalization. The changes across different models under consideration suggest the level of generalization. Since SBF models specify different kinds of structural information (e.g., locations, structural relations, components etc.), successive removal of each kind leads to the formation of models at different levels of abstraction. By removal, we mean two things: (i) substitution of specific values (e.g., low and medium) by a value from a more general class of values (e.g., qualitative-value) in a value hierarchy; and (ii) a complete deletion of specific structural information (e.g., deleting the information that some substance *moves* from one location to another). These will become clearer from the example illustrated below.

Since some functions such as that of a `sulfuric acid cooler` can be classified in multiple ways, multiple subtasks of generalization can be performed—generalization over parameter changes and generalization over changes in location. Depending on which generalization is performed on given experiences, different types of abstract models will be formed. However, in some cases, both might be applicable; in such a case of multiple subtasks, generalization occurs to multiple levels. IDEAL applies both methods, when applicable, in a specific order, that is, it generalizes over parameter changes prior to changes in location. Models at intermediate levels of abstraction are models of prototypical devices (similar to design prototypes [Gero, 1990]) such as the model of a heat exchanger that is applicable to both coolers and heaters. Models at still higher levels of abstraction are such as the model of a physical principle "the zeroth law of thermodynamics" or the physical process "heat flow."

Consider the design of a sulfuric acid cooler (Figure 2) and its function of heating water for the purpose of illustrating the methods. The type of this function (i.e., substance-parameter transformation as well as substance-location transformation) suggests two methods for generalization: (i) generalization over substance-parameter transformation (Figure 5) and (ii) generalization over substance-location transformation (Figure 6). The transitions $transition_{2-3}$ in the behavior "cool acid" of NAC (Figure 4(b)) and $transition_{6-7}$ in the behavior "heat water" of SAC (Figure 3(b)) are selected for generalization because they transform parameters of the substance temperature and the substance location.

The application of the method shown in Figure 5 to these two behaviors results in the description of a generalized model as shown in Figure 7, which is the SBF model of a heat exchanger (a prototypical device). Note that the structural information in the behaviors of SAC and NAC is generalized and so are the parametric relations in the corresponding transitions (Figure 7). For instance, the specific components H_2SO_4-pipe and HNO_3-pipe

```
Input:  • $E_1$, the new design experience.
        • $E_2$, a design experience found to be similar to $E_1$ under the same node in memory.
Output: • Generalized model from $E_1$ and $E_2$.
Procedure:
if (function of $E_1$ is substance-parameter-transformation)
  then
    begin
    (1) Get transitions, $TR_1$ and $TR_2$, corresponding to the transformed parameter
        in $E_1$ and $E_2$ respectively.
    (2) Compare the change in parameters in $TR_1$ and $TR_2$ qualitatively.
        if (direction of change is same in $TR_1$ and $TR_2$)
          then generalize over "range" of the parameters;
          else generalize over the direction of change;
    (3) Modify other context in $TR_1$ and $TR_2$ that specifies this parameter. That is,
        if (any "inequalities" exist on the parameter-relations)
          then generalize the inequalities to conditional inequalities;
    (4) Propagate this generalization to other dependent parameters and transitions,
        and then repeat step (3) until all the context is generalized.
    (5) Store the generalized model from $E_1$ and $E_2$.
    end.
```

Figure 5: **A model-based method for generalizing over parameter transformation**

that achieve the function "allow heat" are generalized to the abstract component pipe achieving the same function, which is prototypical of a heat exchanger.

IDEAL's knowledge of components that H_2SO_4-pipe and HNO_3-pipe belong to the class of *pipes* helps in doing this generalization. Also, the parametric relations in Figure 7 cover both possibilities, that is, *increase* and *decrease* in the substance temperature, unlike those in the behavior of either SAC or NAC alone. This is essential to describing the behavior of a heat exchanger. Further, the generalizations are propagated to the behaviors of those substances on which the transitions depend, which is indicated by UNDER-CONDITION-TRANSITION in the Figures 3(b) & 4(b). That is, in step 4 of the method (Figure 5), for instance, the generalizations performed on the behavior segment (say, "heat water" of sulfuric acid cooler) are propagated to the dependent transition (i.e., "cool acid" of sulfuric acid cooler) which results in the generalized segment "cool substance" shown in Figure 7.

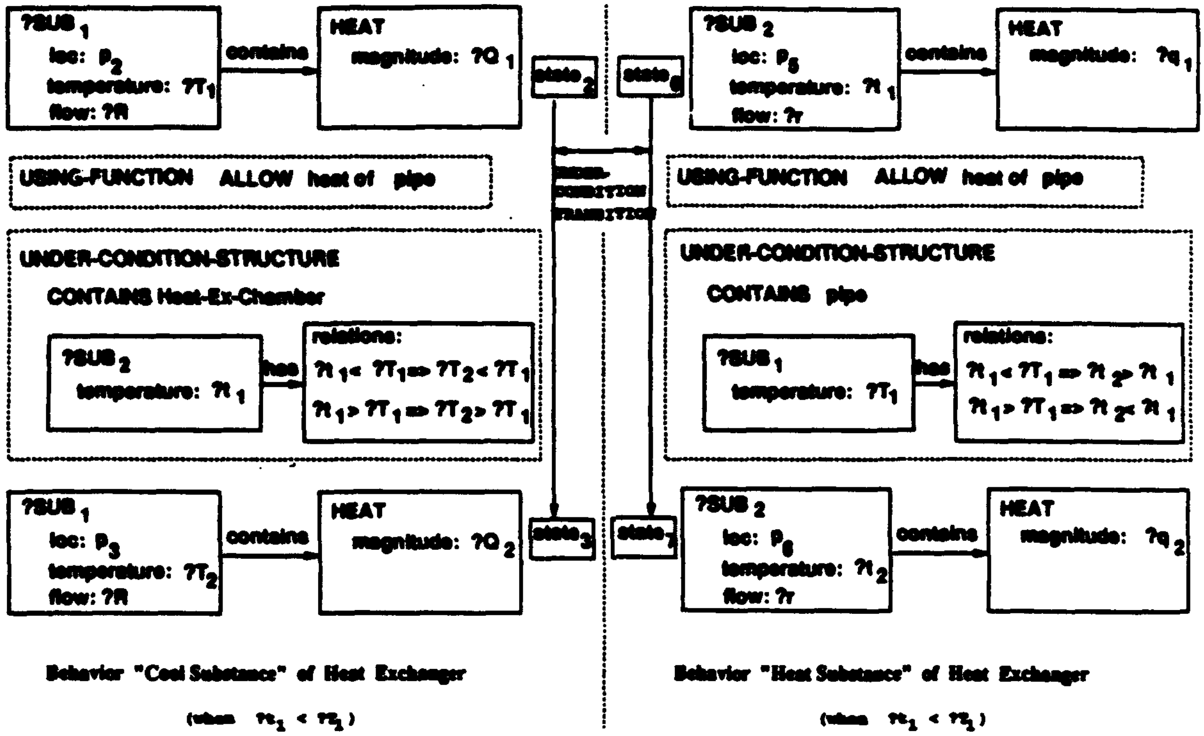
The application of the method shown in Figure 6 to the result of applying the first

```
Input:   • $E_1$, the new design experience or newly generalized experience.
         • $E_2$, a design experience (perhaps generalized before) found to be similar to $E_1$,
           if any, under the same node in memory.
Output:  • Generalized model from $E_1$ (and $E_2$).
Procedure:
if (function of $E_1$ is substance-location-transformation)
  then
    begin
    (1) Get transitions, $TR_1$ and $TR_2$, corresponding to the location in $E_1$and $E_2$ respectively.
    (2) Compare the causal context that involves location in $TR_1$ and $TR_2$.
        if (causal context is similar in $TR_1$ and $TR_2$)
          then generalize/variablize locations;
          else  generalize over the associated structural elements;
    (3) Modify other context that involves locations and associated structural information. That is,
        if (any structural conditions exist in $TR_1$ and $TR_2$ and they are similar)
          then remove the structural conditions;
          else  check for similarity at a more abstract level of components involved;
    (4) Propagate this generalization to other dependent parameters and transitions,
        and then repeat step (3) until all the context is generalized.
    (5) Store the generalized model from $E_1$ and $E_2$.
    end.
```

Figure 6: **A model-based method for generalizing over location transformation**

method, that is, to the model of the heat exchanger, leads to the formation of an even further generalized model as shown in Figure 8. This is the generic principle that we call the zeroth law of thermodynamics. This model is also a description, although partial, of the process "heat flow" because the process of heat flow is the behavior that the zeroth law of thermodynamics epitomizes.[3] However, the system, conforming to the classical "term problem" in learning, does not realize that this is *the* zeroth law of thermodynamics nor does it realize that this is a partial description of the process "heat flow," but rather considers it simply as an abstract model possibly applicable to a wider class of devices. Again, note that the structural information in the behavior of heat exchanger is further generalized in the zeroth law of thermodynamics. For instance, the component pipe that achieves the function "allow heat" is generalized to an abstract component connector achieving the same function.

[3] A complete description should also indicate that heat continues to flow from a hot body to a cold body only until an equilibrium temperature is reached.



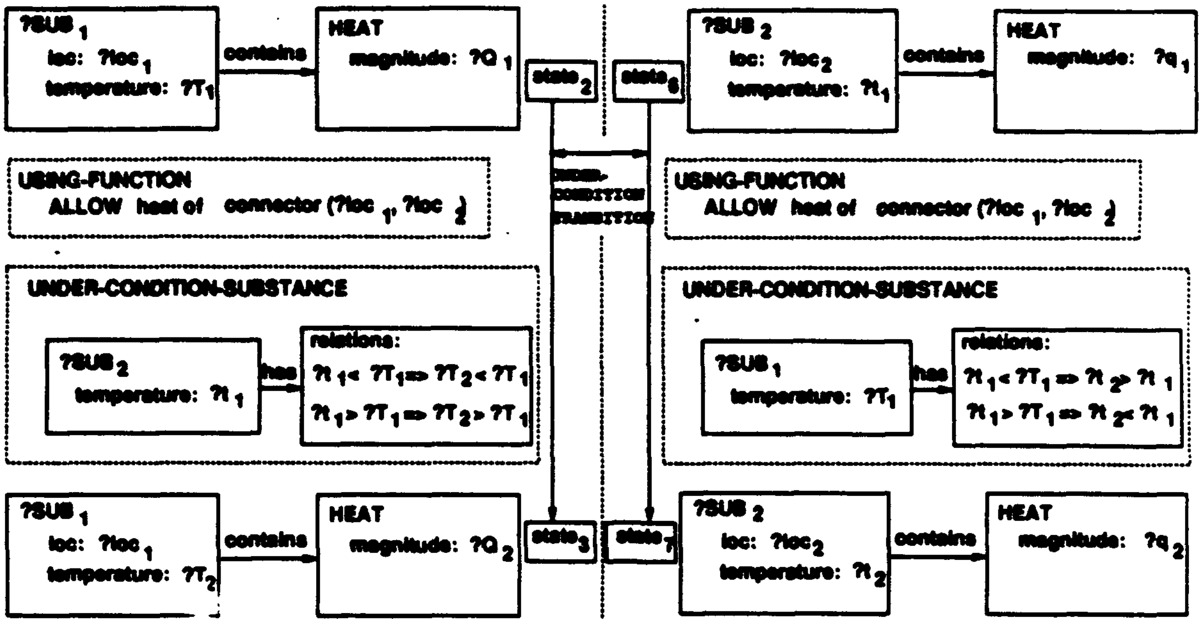
Note: Symbols prefixed with ? denote variables.
All locations are with reference to components in this design.
All labels for states and transitions are local to this design.

Figure 7: **Behavior of a Heat Exchanger generalized from SAC and NAC**

In addition to the result of generalization over structure, the generalized parametric relations in Figure 8 that cover both *increase* and *decrease* in the substance temperature are also crucial to representing the zeroth law of thermodynamics. These relations are represented as conditions on substance properties indicated by the annotation UNDER-CONDITION-SUBSTANCE in Figure 8 because the structural conditions (in Figure 7) are removed by the application of step 3 in Figure 6. Again, step 4 in the method shown in Figure 6 leads to the propagation of generalizations performed in one behavioral segment to the dependent ones.

4.1 Learning Indices to Hypothesized Models

When a piece of knowledge is learned, its usefulness relies on the ability to also learn the appropriate conditions under which it might be used. In other words, learning a piece of knowledge inevitably involves learning its *indices*. So, when the models of the heat exchanger



Note: Symbols prefixed with ? denote variables.
All locations are with reference to components in this design.
All labels for states and transitions are local to this design.

Figure 8: **The Zeroth Law of Thermodynamics**

and the zeroth law of thermodynamics are hypothesized, one subtask is to learn their indices. Since these models are learned in the context of analogical design and are intended for the design task, like design cases, they could also be indexed by different types of indices—functional and structural.

Storing the hypothesized models in a hierarchically organized memory implies two distinct issues in index learning: *learning the indexing vocabulary* and *learning the right level of generalization* [Bhatta and Goel, 1992]. Deciding on the indexing vocabulary generally requires some notion of what is important about the new knowledge and the task for which it is likely to be reused. The level of generalization depends in part on the knowledge already stored in memory and the inductive biases that can be generated at storage time.

We have earlier shown how the SBF model of a design, together with a specification of the task for which the design case might be reused, provides the vocabulary for indexing the design case in memory [Bhatta and Goel, 1992] . Further, we have also shown how the model-based method, together with similarity-based learning (using earlier design cases in memory) helps to determine the level of index generalization. Insofar as the same types of indices are used for storing the models of heat exchangers and the models of the zeroth law of thermodynamics, the same methods as presented in [Bhatta and Goel, 1992] apply to the task of learning indices to the hypothesized models. The only difference is that the indices for these models will be more general than those for design cases. Hence these models will be stored at a more general level in a hierarchical organization of memory. However, space constraints do not permit us to describe these methods here.

5 Evaluation

The proposed model-based method can be evaluated for different things: (i) Computational Efficiency; (ii) Domain Generality; and (iii) Performance Task.

Computational Efficiency: The issue here is whether IDEAL requires a number of examples in order to learn a target concept such as the zeroth law of thermodynamics. Due to the constraints that SBF models provide on the generalization process, IDEAL does not require more than a few (e.g., 3-4) examples for learning the zeroth law of thermodynamics. From the examples illustrated in this paper, it is clear that IDEAL required only two examples for learning a reasonably complete description of the zeroth law of thermodynamics.

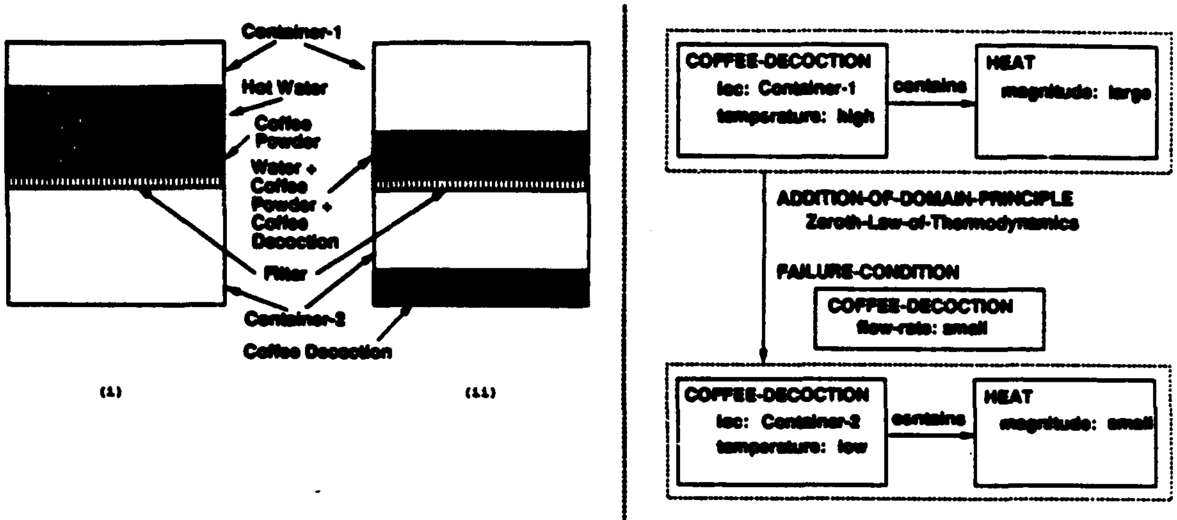
Domain Generality: The question here is whether the proposed methods are applicable in different domains? Currently, they have been tested only in the domain of heat exchangers; that is, for learning the model of a heat exchanger and the zeroth law of thermodynamics. However, from our experience with other tasks for which SBF models have been used, such as case-based design [Goel, 1989; Goel, 1991a; Goel, 1991b] and index learning by model-based generalization [Bhatta and Goel, 1992; Bhatta and Goel, 1993], in different domains, it appears that the proposed method is also applicable in other domains of physical devices such as electric circuits (i.e., for instance, in learning Ohm's law). This is because the main power of the method comes from the representational framework that the SBF models provide.

Performance Task: The question here is whether model-based learning of principles or processes affect some performance task. The motivation is that a target concept is best learned if done in the context of a performance task in which it gets used. We consider two related but different performance tasks, namely, device redesign and design of physical devices by analogy in which learned principles are useful.

(i) **Device Redesign:** The device redesign task takes as input a failed design and the feedback from the environment in which the device operates, and gives as output a new modified design. The physical principles learned by IDEAL such as the zeroth law of thermodynamics are useful in device redesign. For instance, Prabhakar and Goel [1992] have described how the zeroth law of thermodynamics (similar to the representation learned by IDEAL) is useful in redesigning a failed coffee maker. Device redesign task in their work involves four subtasks: formation of causal explanations of failures, discovery of new design constraints, formulation of internal behaviors that accommodate the modified constraints, and redesign of the device structure for realizing the modified internal behaviors. In particular, they describe how the zeroth law of thermodynamics is useful in the formation of causal explanations of why the device failed.

Consider, for instance, the design of a simple coffee maker whose structure is shown in Figure 9(a) in two states ((i)before and (ii) after coffee decoction is formed in `Container-2`). Its function of making coffee is to produce coffee decoction in `Container-2`, given hot water and coffee powder in `Container-1`. This design satisfies the function desired of the coffee maker, but its behavior is suboptimal. That is, there are two problems: (i) coffee decoction formed in `Container-2` is only lukewarm, and (ii) it does not stay warm in `Container-2`.

The first subtask in redesigning this coffee maker is to form a causal explanation for the failure. One way to accomplish this task is by instantiating an abstract principle, such as the zeroth law of thermodynamics, in the context of the current design and its environment. The zeroth law of thermodynamics can be accessed by using an abstraction of the failure in the coffee maker (which is "loss of heat to the environment") as a probe into memory. Instantiating the law in the context of the design of coffee maker results in the SBF model for the failure behavior of coffee maker as shown in Figure 9(b). This helps in formulat-



(a) Structure of A Simple Coffee Maker (in two states)

(b) SBF Model for Failure Behavior of the Coffee Maker

Figure 9: Redesigning A Simple Coffee Maker

ing new design constraints and in solving the subsequent subtasks in the redesign. The redesigned coffee maker may have a plunger in `Container-1` to pump the decoction into `Container-2` and a hot plate beneath `Container-2` to keep the decoction in `Container-2` warm (see [Prabhakar and Goel, 1992] for details).

(ii) Analogical Design: The hypothesized abstract models, that is, the models of the heat exchanger and the zeroth law of thermodynamics, can also be tested by using them for design by analogy. In cross-domain analogical transfer, the applicability of a source analog to a target problem is often due to sharing of some high-level (abstract) principles governing both the source and the target domains. In IDEAL, we are currently exploring the role of prototypical devices and physical principles in cross-domain transfer. Transfer in this scheme involves accessing the abstractions associated with a source analog and then instantiating them in the target domain rather than directly mapping source-specific substructures onto the target problem. We call this method *model-based analogy.* (See [Bhatta, 1992] for details). The models of physical principles and processes are more abstract than the models of prototypical devices. Hence, physical principles and processes are applicable to a wider class of design problems and thus they facilitate analogical transfer between distant domains.

6 Related Work

Learning Task: Our work is similar to JULIANA [Shinn, 1989] and ASIS [Roverso *et al.*, 1992] in learning abstractions from specific cases. But our learning task is different in the type of abstractions learned: JULIANA forms abstract cases and ASIS forms abstractions of structural models of specific situations while IDEAL discovers models of prototypical devices. physical principles, and processes.

Explanations in Learning: The proposal that learning from experience is facilitated by explanations of specific experiences dates at least as far back as Winston [1980]. Winston's model assumed knowledge of "what" is the concept being learned and relied on information concerning whether an example is a positive instance or negative instance of the concept. Our approach is similar to Winston's later models [1982; 1986] that show that learning can be done by analogically transferring causal links in the explanation of an example to the target concept.

Our approach is also similar to explanation-based learning (EBL) [DeJong and Mooney, 1986; Mitchell *et al.*, 1986] in using explanations (SBF models) to constrain the learning of "concepts." However, most EBL systems assume some knowledge of the target concept *a priori;* our model-based approach attempts to discover them.

Also, our model-based approach differs from EBL in the kind of explanations it uses. First, while the explanations in EBL are purely causal, the explanations in SBF models are functional in nature, i.e., they not only provide a causal account, they also show how causal processes result in the achievement of specific functions. Second, the explanations in EBL specify how an example is an instance of a target concept while SBF models are explanations of the functioning of devices. Besides, models also provide functional and structural decomposition knowledge for the devices that is useful in constraining the generalization process. Third, the explanations in EBL are constructed at run-time from domain specific rules whereas SBF models are formed by revising old models as part of the problem solving. Fourth, SBF models are grounded in a well-defined component-substance ontology.

Integration of Learning Methods: In addition, our work integrates the model-based approach with similarity-based methods for learning abstractions. In this respect, our work is similar to Pazzani's OCCAM [1991] which integrates similarity-based learning, EBL, and theory-driven learning (TDL) for learning of concepts.

Learning by Discovery: Our approach can be compared to work in scientific discovery such as BACON [Langley *et al.*, 1987], FAHRENHEIT [Zytkow, 1987], and ABACUS [Falkenhainer and Michalski, 1986]. These systems require a large amount of data because they use inductive approaches to discover regularities and form laws. In contrast, IDEAL is designed to incrementally discover physical principles using models to guide the discovery process. Hence, we expect IDEAL to require fewer examples for discovering useful principles. Most of the above systems use predesigned experiments to test their hypotheses. On the other hand. ur approach takes a different stance on experimentation—it views *problem solving* using hypothesized "concepts" as testing the hypotheses. Thus hypothesis testing is not planned but rather is a consequence of solving design problems in the real world.

7 Conclusions

The models of specific devices (SBF models) provide both the content and the constraints for learning the models of physical principles (BF models) by incremental generalization over design experiences. In particular, we showed that the function of a device determines what to generalize from its SBF model, the SBF model suggests how far to generalize, and the typology of functions indicates what method to use for generalization.

Without the constraints from models (or similar knowledge) the method of induction for generalization can be potentially very complex. So the moral is that the existing machine learning techniques can be adapted for learning design knowledge, but they may need to be constrained by deep knowledge such as models in order to circumvent the complexity problem. Furthermore, most existing machine learning techniques have been developed in isolation of a performance task, but we believe that the acquisition of knowledge cannot be separated from the problem-solving tasks in which the learned knowledge might be used.

Finally, we believe that the issue of learning abstract models such as the models of physical principles and processes that facilitate cross-domain analogical design provides a great potential for machine learning in design because cross-domain analogies often play a crucial role in non-routine design.

Acknowledgements. This research has benefited from discussions with Eleni Stroulia. This work has been supported by research grants from NSF (grant IRI-92-10925), ONR (contract N00014-92-J-1234), Northern Telecom, ARPA, and equipment grants and donations from IBM, NCR..and Symbolics. We thank Kavi Mahesh and anonymous reviewers for their comments on an earlier draft of this paper.

References

[Alterman, 1988] R. Alterman. Adaptive Planning. *Cognitive Science*, 12:393–422, 1988.

[Bhatta and Goel, 1992] S. Bhatta and A. Goel. Use of Mental Models for Constraining Index Learning in Experience-Based Design. In *Proceedings of the AAAI workshop on Constraining Learning with Prior Knowledge*, pages 1–10, San Jose, CA, July 1992.

[Bhatta and Goel, 1993] S. Bhatta and A. Goel. Model-Based Learning of Structural Indices to Design Cases. In *Proceedings of the IJCAI workshop on "Reuse of Designs: An Interdisciplinary Cognitive Approach"*, pages A1–A13, Chambery, Savoie, France, August 1993.

[Bhatta, 1992] S. Bhatta. A Model-Based Approach to Analogical Reasoning and Learning in Design. Technical Report GIT-CC-92/60, Georgia Institute of Technology, College of Computing. Atlanta, GA, November 1992. Ph.D. Thesis Proposal.

[Bylander and Chandrasekaran, 1985] T. Bylander and B. Chandrasekaran. Understanding Behavior Using Consolidation. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 450–454, 1985.

[DeJong and Mooney, 1986] G. DeJong and R. Mooney. Explanation-Based Learning: An Alternative View. *Machine Learning*, 1(2):145–176, 1986.

[Falkenhainer and Michalski, 1986] B. Falkenhainer and R. Michalski. Integrating Quantitative and Qualitative Discovery: The ABACUS System. *Machine Learning*, 1:367–401, 1986.

[Fermi, 1937] E. Fermi. *Thermodynamics*. Prentice-Hall, New York, NY, 1937.

[Gentner, 1983] D. Gentner. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7:155–170, 1983.

[Gero, 1990] J. Gero. Design Prototypes: A Knowledge Representation Schema for Design. *AI Magazine*, 11(4):26–36, 1990.

[Goel, 1989] A. Goel. *Integration of Case-Based Reasoning and Model-Based Reasoning for Adaptive Design Problem Solving*. Ph.D. thesis, The Ohio State University, Department of Computer and Information Science, Columbus, Ohio, 1989.

[Goel, 1991a] A. Goel. A Model-Based Approach to Case Adaptation. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 143–148, Chicago, August 1991.

[Goel, 1991b] A. Goel. Model Revision: A Theory of Incremental Model Learning. In *Proceedings of the Eighth International Conference on Machine Learning*, pages 605–609, Chicago, June 1991.

[Goel, 1992] A. Goel. Representation of Design Functions in Experience-Based Design. In D. Brown, M. Waldron, and H. Yoshikawa, editors, *Intelligent Computer Aided Design*, pages 283–308. North-Holland, Amsterdam, Netherlands, 1992.

[Hammond, 1989] K. Hammond. *Case-Based Planning: Viewing Planning as a Memory Task.* Academic Press, Boston, MA, 1989.

[Kerr and Duffy, 1992] S. Kerr and A. Duffy. Dynamic Memory by Automating Rationalization of Past Designs. In *Proceedings of the AID'92 Workshop on Machine Learning in Design*, Pittsburgh, PA, USA, June 1992.

[Klahr and Dunbar, 1988] D. Klahr and K. Dunbar. Dual Space Search During Scientific Reasoning. *Cognitive Science*, 12:1–48, 1988.

[Kolodner and Simpson, 1989] J. L. Kolodner and R. L. Simpson. The MEDIATOR: Analysis of An Early Case-Based Probel Solver. *Cognitive Science*, 13(4):507–549, 1989.

[Langley *et al.*, 1987] P. Langley, H. Simon, G. Bradshaw, and J. Zytkow. *Scientific Discovery: An Account of the Creative Processes.* MIT Press, Boston, MA, 1987.

[Mitchell *et al.*, 1986] T. M. Mitchell. R. Keller, and S. Kedar-Cabelli. Explanation-Based Generalization: A Unifying View. *Machine Learning*, 1(1):47–80, 1986.

[Pazzani, 1991] M. Pazzani. Learning to Predict and Explain: An Integration of Similarity-Based, Theory-Driven, and Explanation-Based Learning. *The Journal of the Learning Sciences*, 1(2):153–199, 1991.

[Prabhakar and Goel, 1992] S. Prabhakar and A. Goel. Integrating Case-Based and Model-Based Reasoning for Creative Design: Constraint Discovery, Model Revision, and Case Composition. In *Proceedings of the Second International Conference on Computational Models of Creative Design*, Heron Island, Australia, December 1992.

[Riesbeck and Schank, 1989] C. Riesbeck and R. Schank. *Inside Case-Based Reasoning*. Erlbaum, Hillsdale, NJ, 1989.

[Rissland and Ashley, 1987] E. Rissland and K. Ashley. HYPO: A Case-Based Reasoning System. In *Proceedings of Tenth International Joint Conference on Artificial Intelligence*, 1987.

[Roverso *et al.*, 1992] D. Roverso, P. Edwards, and D. Sleeman. Machine Discovery by Model Driven Analogy. In J. M. Zytkow, editor. *Proceedings of the ML-92 workshop on Machine Discovery*, pages 87–97, Aberdeen, Scotland, July 1992.

[Sembugamoorthy and Chandrasekaran, 1986] V. Sembugamoorthy and B. Chandrasekaran. Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems. In J.Kolodner and C.Riesbeck, editors, *Experience, Memory and Reasoning*, pages 47–73. Lawrence Erlbaum, Hillsdale, NJ, 1986.

[Shinn, 1989] H. S. Shinn. *A Unified Approach to Analogical Reasoning.* Ph.D. thesis, Georgia Institute of Technology, School of Information and Computer Science, Atlanta, GA, 1989.

[Winston, 1980] P. Winston. Learning and Reasoning by Analogy. *Communications of the ACM*, 23(12), 1980.

[Winston, 1982] P. Winston. Learning New Principles from Precedents and Exercises. *Artificial Intelligence*, 19(3):321–350, 1982.

[Winston, 1986] P. Winston. Learning by Augmenting Rules and Accumulating Censors. In R.Michalski, J.Carbonell, and T.Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach, Vol. II.* Morgan Kaufmann, Los Altos, CA, 1986.

[Zytkow, 1987] J. Zytkow. Combining Many Searches in the FAHRENHEIT Discovery System. In *Proceedings of the Fourth International Workshop on Machine Learning*, pages 281–287. Los Altos, CA, 1987. Morgan Kaufmann.

Learning Generic Mechanisms from Experiences for Analogical Reasoning[*]

Sambasiva R. Bhatta and **Ashok K. Goel**
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
{bhatta,goel}@cc.gatech.edu

Abstract

Humans appear to often solve problems in a new domain by transferring their expertise from a more familiar domain. However, making such cross-domain analogies is hard and often requires abstractions common to the source and target domains. Recent work in case-based design suggests that generic mechanisms are one type of abstractions used by designers. However, one important yet unexplored issue is where these generic mechanisms come from. We hypothesize that they are acquired incrementally from problem-solving experiences in familiar domains by generalization over patterns of regularity. Three important issues in generalization from experiences are what to generalize from an experience, how far to generalize, and what methods to use. In this paper, we show that mental models in a familiar domain provide the content, and together with the problem-solving context in which learning occurs, also provide the constraints for learning generic mechanisms from design experiences. In particular, we show how the model-based learning method integrated with similarity-based learning addresses the issues in generalization from experiences.

Introduction

Analogy is often believed to play an important role in reasoning underlying innovation and creativity. Analogies can be of different types: within-problem, cross-problem but within-domain, and cross-domain. We are interested in studying cross-domain analogies. Psychological research shows that humans use abstractions in making cross-domain analogies (e.g., Gick & Holyoak, 1983; Catrambone & Holyoak, 1989). Some of the issues of interest then are how reasoning is mediated by the abstractions (shared between the source and target domains) and how those abstractions are learned. We explore the latter issue in the context of the design of physical devices such as electric circuits and heat exchangers. Our goal is to build a computational model that can account for these phenomena and use it to generate testable predictions about designers' behavior.

Goel (1989) has proposed models of generic teleological mechanisms (GTMs), such as cascading, feedback, and feedforward, as one type of abstract knowledge that designers use in case-based design. GTMs take as input the functions of a desired design and a known design, and suggest patterned modifications to the structure of the known design that would result in the desired design. Stroulia and Goel (1992) have shown that GTMs indeed are useful in non-routine adaptive design. But one important yet unexplored issue is how these GTMs are acquired. Our hypothesis is that they are acquired incrementally from problem-solving experiences in familiar domains by generalization over patterns of regularity. For instance, a designer may acquire from examples in the domain of electric circuits a model of cascading, and when and how to cascade a number of similar components together (i.e., to connect multiple components to amplify the overall delivered function). The designer can then use that model for designing in a different domain such as the domain of heat exchangers.

Generalization from experiences raises three important issues. First is the issue of relevance, that is, the issue of deciding what to generalize from an experience. We represent in design experiences a designer's comprehension of how devices work (i.e., how the structure of a design results in its output behaviors). We represent this comprehension as structure-behavior-function (SBF) models and represent the models of GTMs as behavior-function (BF) models. We propose that the problem-solving context in which learning occurs together with the hierarchical organization of the SBF model of the device help determine what to generalize from the model. Further, the SBF models lead to a typology of behavioral patterns over which the generalization process can result in learning GTMs. Second, how far a chosen aspect of the device can be generalized. We show that the similarities in the SBF models of the cur-

[*]This work has been supported by research grants from ONR (contract N00014-92-J-1234), NSF (grant C36-688), Northern Telecom, Georgia Tech Research Corporation, and a CER grant from NSF (grant CCR-86-19886), and equipment grants and donations from IBM, Symbolics, and NCR.

In the Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society, Boulder, CO. June 18-21, 1993.

rent design and related designs in a case memory can help determine how far to generalize. Third. what methods can be used for generalization. We show that a typology of the patterns of regularity in SBF models can help to determine what strategy to use.

The system IDEAL[1] implements the proposed learning method. We evaluate the learning method by showing how the GTMs learned in one domain can facilitate designing in another domain.

The Learning Task

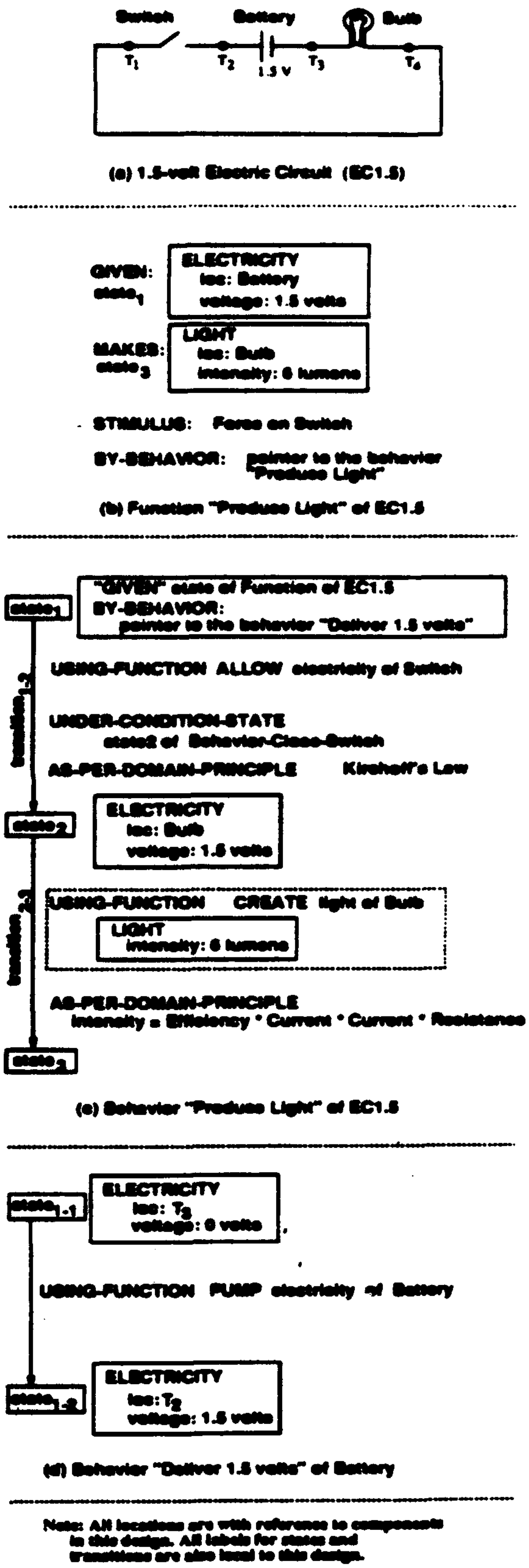
The Problem-Solving Context: IDEAL takes as input a specification of the functional and structural constraints on a desired design. and gives as output a structure that realizes the specified function and satisfies the structural constraints: it also gives an SBF model that explains how the structure realizes that function. A design case in IDEAL specifies (i) the functions delivered by the stored design (ii) the structure of the design. and (iii) a p· .ter to the causal behaviors of the design (the SBF model). IDEAL indexes its design cases both by functions that the stored designs deliver and by the structural constraints they satisfy.

IDEAL's **learning task** takes as input a design experience and forms the BF model of a GTM. The input knowledge structure for the learning task is the case-specific SBF model of the given design experience and the output knowledge structure is the case-independent BF model of a GTM. The learned GTM is such that it is an abstraction over certain patterns of regularity (explained later) observed in the structure and behavior of the given SBF model and the model of the most similar experience in case memory.

Case-Specific SBF Models

IDEAL's models of specific devices are represented in the form of structure-behavior-function (SBF) models. These models are based on a *component-substance ontology* (Bylander. 1991). In this ontology, the structure of a device is viewed as constituted of *components* and *substances*. Substances have *locations* in reference to the components in the device. They also have *behavioral properties*, such as *voltage* of *electricity*, and corresponding *parameters*, such as *1.5 volts*, *3 volts*, etc. This ontology gives rise to a *behavioral representation language* (Goel, 1989) for describing the SBF model of a design that is a generalization on functional representation scheme (Sembugamoorthy & Chandrasekaran. 1986: Chandrasekaran. Goel. & Iwasaki. 1993). The constituents of the SBF model are described below.

Structure: The structure of a design is expressed in terms of its constituent components and substances and the interactions between


Switch
Battery
Bulb
T1
T2
1.5 V
T3
T4
GIVEN:
state1
ELECTRICITY
loc: Battery
voltage: 1.5 volts
MAKES:
state3
LIGHT
loc: Bulb
intensity: 6 lumens
STIMULUS: Force on Switch
BY-BEHAVIOR: pointer to the behavior "Produce Light"
state1
"GIVEN" state of Function of EC1.5
BY-BEHAVIOR:
pointer to the behavior "Deliver 1.5 volts"
transition1-2
USING-FUNCTION ALLOW electricity of Switch
UNDER-CONDITION-STATE
state2 of Behavior-Close-Switch
AS-PER-DOMAIN-PRINCIPLE Kirchoff's Law
state2
ELECTRICITY
loc: Bulb
voltage: 1.5 volts
transition2-3
USING-FUNCTION CREATE light of Bulb
LIGHT
intensity: 6 lumens
AS-PER-DOMAIN-PRINCIPLE
intensity = Efficiency * Current * Current * Resistance
state3
state1-1
ELECTRICITY
loc: T3
voltage: 0 volts
USING-FUNCTION PUMP electricity of Battery
state1-2
ELECTRICITY
loc: T2
voltage: 1.5 volts

(a) 1.5-volt Electric Circuit (EC1.5)

(b) Function "Produce Light" of EC1.5

(c) Behavior "Produce Light" of EC1.5

(d) Behavior "Deliver 1.5 volts" of Battery

Note: All locations are with reference to components in this design. All labels for states and transitions are also local to this design.

Figure 1: Design of A 1.5-volt Electric Circuit (EC1.5)

[1] IDEAL stands for Integrated "DEsign by Analogy and Learning."

them. Figure 1(a) shows the structure of a **1.5-volt electric circuit** (EC1.5) schematically.

Function: A function is represented as a schema that specifies the behavioral state the function takes as input, the behavioral state it gives as output, and a pointer to the internal causal behavior of the design that achieves the function. Figure 1(b) shows the function "Produce Light" of EC1.5. Both the input state and the output state are represented as *substance schemas*. The input state specifies that electricity at **location** Battery in the topography of the device (Figure 1(a)) has the property **voltage** and the corresponding parameter **1.5 volts**. The output state specifies the property **intensity** and the corresponding parameter **6 lumens** of a different substance, light, at **location** Bulb. In addition, the slot *by-behavior* acts as an index into the causal behavior that achieves the function of producing light.

Behavior: The internal causal behaviors of a device are viewed as sequences of *state transitions* between *behavioral states*. The annotations on the state transitions express the *causal*, *structural*, and *functional context* in which the transformation of state variables, such as substance, location, properties, and parameters, can occur. Figure 1(c) shows the causal behavior that explains how electricity in Battery is transformed into light in Bulb. *State*$_2$ is the preceding state of *transition*$_{2-3}$ and *state*$_3$ is its succeeding state. *State*$_1$ describes the state of electricity at location Battery and so does *state*$_2$ at location Bulb. *State*$_3$ however describes the state of light at location Bulb. The annotation USING-FUNCTION in *transition*$_{2-3}$ indicates that the transition occurs due to the primitive function "create light" of Bulb.

The causal behaviors can be specified at different levels of detail. For instance, *state*$_1$ is an aggregation of a sequence of several states and state transitions at a different level as shown in Figure 1(d).

Case-Independent BF Models

Generic Teleological Mechanisms (GTMs) are one type of knowledge that designers use in adaptive design, that is, in modifying an old design by insertion of specific patterns of components (or substructures) (Stroulia & Goel, 1992). Examples of GTMs are cascading, feedback, and feedforward. GTMs are *teleological* because they result in specific functions and are *generic* because they are case independent. For example, the cascading mechanism takes as input the desired function and the function (with a lesser range) delivered by an available device, and suggests a structural pattern (i.e., the replication) of the available device that delivers the desired function. Further, the cascading mechanism can be instantiated in any specific device that satisfies its applicability conditions. For instance, one applicability condition is that the functions delivered by each replicated device should add up to give the desired function (i.e., the replication should be *functionally additive*). More precisely, the condition is that the smaller parametric transformation delivered by each replicated device should sum up to provide the desired larger transformation.

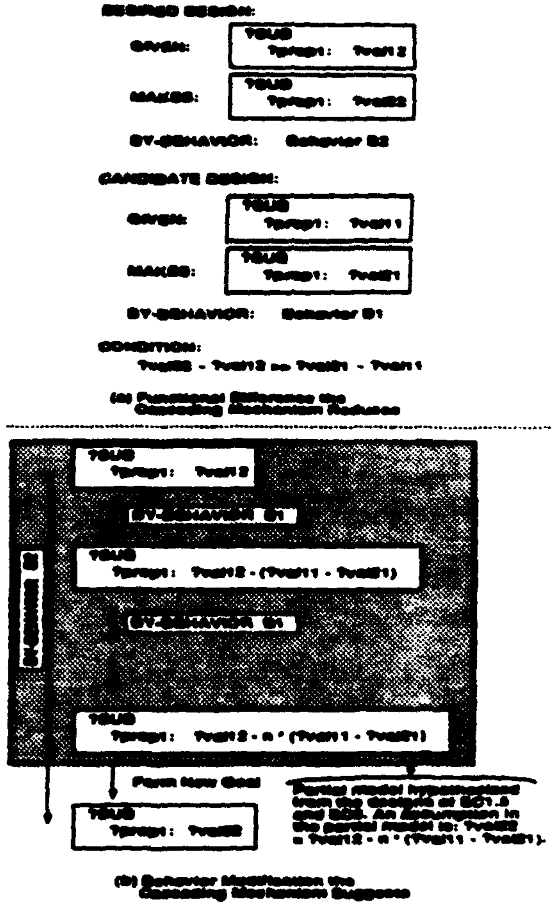
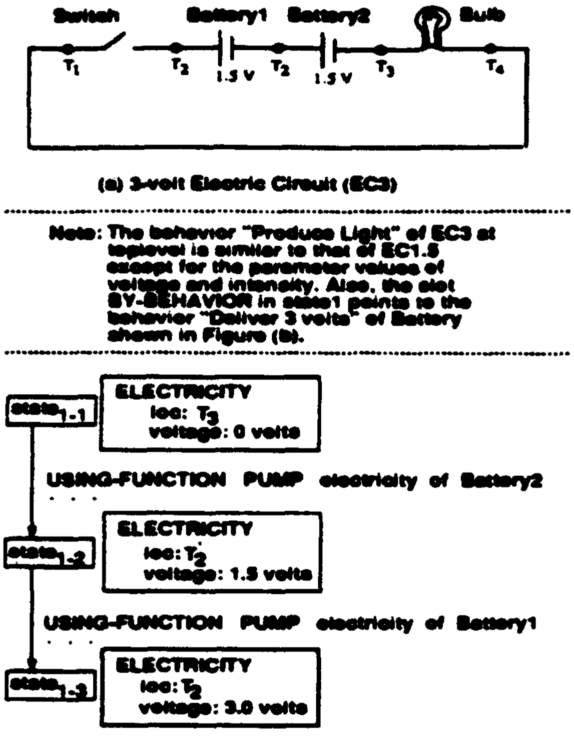

DESIRED DESIGN:
GIVEN:
MAKES:
BY-BEHAVIOR:
CANDIDATE DESIGN:
GIVEN:
MAKES:
BY-BEHAVIOR:
CONDITION:
(a) Functional Difference the Cascading Mechanism Reduces
Form New Goal
(b) Behavior Modification the Cascading Mechanism Suggests

Figure 2: **BF Model of the Cascading Mechanism**

The BF model representation of a GTM encapsulates two types of knowledge: knowledge about the difference between the functions of a known design and a desired design that the GTM can help reduce; and knowledge about modifications to the internal causal behaviors of the known design that are necessary to reduce this difference. For example, Figures 2(a) & 2(b) respectively show these two types of knowledge for the cascading mechanism. The model of cascading indicates that a behavior can be replicated as much as possible to achieve a desired function and finally a goal be formed to find a component that can deliver the residual function. This additional component is needed when the desired function is not an integral multiple of the function of each replicated device.

The Learning Method

Suppose, for instance, IDEAL's case memory has the design of EC1.5 shown in Figure 1. Let us



(a) 3-volt Electric Circuit (EC3)

Note: The behavior "Produce Light" of EC3 at toplevel is similar to that of EC1.5 except for the parameter values of voltage and intensity. Also, the slot BY-BEHAVIOR in state1 points to the behavior "Deliver 3 volts" of Battery shown in Figure (b).

(b) Behavior "Deliver 3 volts" of Battery

Figure 3: **Design of A 3-volt Electric Circuit (EC3)**

now consider the scenario where IDEAL is presented with a problem of designing a 3-volt electric circuit (EC3) that delivers the function "produce light of intensity 12 lumens in the bulb when the switch is closed, given that there is electricity with a voltage of 3 volts in the battery" and satisfies the structural constraint "the design cannot have a single 3-volt battery." IDEAL retrieves the design case EC1.5 because the given functional specification is similar to the function of EC1.5. However, IDEAL may know only how to replace a component in a past design to solve the current problem. The component-replacement plan specifies how to replace the component that is responsible for the functional difference by a new component that reduces the functional difference and thus enables the overall device to deliver the desired function. In such cases, IDEAL fails to solve the current problem due to the structural constraint specified. Then, if an oracle presents the correct solution that both delivers the desired function and satisfies the structural constraint (the schematic of the structure of the new device is shown in Figure 3(a)), IDEAL learns how the new device behaves (a segment is shown in Figure 3(b)) by revising the behavior of EC1.5. This problem-solving context enables IDEAL to focus on the substructure that delivers the required voltage for comparing with the corresponding substructure in the old case EC1.5. By generalizing over the structural pattern (in this substructure) and the corresponding behavioral segments, it learns the cascading mechanism. We will now focus on the learning of the cascading mechanism.

The learning method is model-based in that the SBF models of the design cases provide the content for generalizing over the patterns of regularity in the device structure and device behavior. The representation vocabulary of the SBF models further leads to several classes of regularity, a few of which that are relevant to learning cascading mechanism are: (i) repetition of behavioral segments, that is, a sequence of state-transitions repeats several (say, *n*) times in the overall device behavior: since a behavior typically corresponds to a structural part (i.e., a component), the corresponding structural regularity is the repetition of the structural part; (ii) repetition of a range of input-output transformation, that is, the same amount of parameter transformation repeats several (say, *n*) times in the device behavior. The two variables of interest for generalization then are the *range of transformation* (*r*) and *number of repetitions of same structure* (*n*). Given the task of learning from two design cases and that there are two variables, four different situations are possible as shown in Table 1. In this paper we will focus on situation **2** only.

The learning method first traverses each focused behavior in the given two designs to notice the above types of regularities, in particular, to identify the values for *n* and *r*. Then it compares the values for the two variables in both the designs and generalizes over them if any similarity exists. The first step of the learning method can be facilitated by indexing from the component into the behavioral segments in which some function of the component plays a role.

In the above problem-solving scenario, the problem-solving context indicates that the behavioral segments to focus on for learning are those that correspond to the function of **Battery** in the two designs, EC1.5 and EC3. Applying the above learning method, it is easy to identify that the learning situation here is **2** shown in Table 1. Generalizing over the number of repetitions and variablizing the range of parameter transformation, IDEAL hypothesizes a GTM that would help in a problem-solving context similar to the current one. The model of the learned (more precisely, *hypothesized*) cascading mechanism and its index are shown in Figure 2 (representations in (a) and the shaded region of (b)); the functional difference that the cascading mechanism reduces is the index for the mechanism.[2]

IDEAL can revise the hypothesized model into a more complete one when it solves a new design problem whose solution has a *structural pattern* that is an instance of the complete cascading mechanism. Thus acquiring a complete model of the cascading mechanism may involve solving a number of design problems incrementally.

[2] A new piece of knowledge learned is futile unless its applicability conditions (or *indices* as we call them) are also learned.

Table 1: **Situations of Regularity Between Similar Components in Two Designs**

Situation	Range of Input-Output Transformation in both designs. *r*	Number of Repetitions in both designs. *n*	What can be Learned?
1.	equal	equal	None due to lack of variation.
2.	**equal**	**not equal**	**Generalization over *n*.** **(e.g.. the cascading mechanism)**
3.	not equal	equal	Generalization over *r*. (e.g.. prototypical device models)
4.	not equal	not equal	None due to lack of regularity.

Evaluation

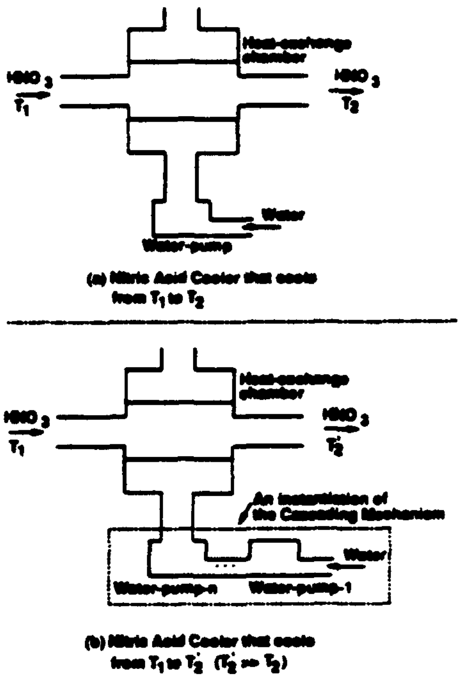
One method for evaluating the learning is to show how the learned mechanisms can affect IDEAL's performance task of designing physical devices. In particular. does it enable IDEAL to transfer design knowledge from one domain (say. electric circuits) to another domain (say. heat exchangers)?

We have tested IDEAL with several designs from the domain of electric circuits and heat exchangers. In one experiment. we gave IDEAL designs of electric circuits such as those illustrated in this paper. IDEAL learned the mechanism of cascading. indexed it by the applicability conditions of the mechanism. and stored it in its memory. Then we gave IDEAL a design problem in the domain of heat exchangers. This problem. relative to IDEAL's knowledge. was such that in order to solve it IDEAL would need to evoke the cascading mechanism. We observed that IDEAL noticed the difference between the desired function and the function of an available device. It then used the functional difference as a probe into its memory. retrieved the cascading mechanism. and solved the new problem by instantiating the retrieved mechanism. More specifically. Figure 4 illustrates how IDEAL instantiated the cascading mechanism learned from the two designs. EC1.5 and EC3. in the water pumps in designing a nitric acid cooler that provides a higher range of cooling (i.e.. T_1-T_2'). (Stroulia & Goel, 1992) provides more details of the adaptation process.

Together. these experiments indicate the utility and effectiveness of our model-based method for learning GTMs: the SBF models enable learning of GTMs in one domain and the learned BF models of GTMs facilitate designing in another. We are currently testing IDEAL with design problems from other domains such as reaction wheel assemblies. and for other mechanisms such as feedback and feedforward.

Related Work

This work on IDEAL evolves from our earlier work on KRITIK (Goel. 1989). IDEAL's component-substance ontology. SBF models. and behavioral representation language all are borrowed from KRITIK. The problem-solving component of IDEAL evolves from KRITIK2 (Stroulia *et al.*, 1992).



(a) Nitric Acid Cooler that cools from T_1 to T_2

(b) Nitric Acid Cooler that cools from T_1 to T_2' ($T_2' >> T_2$)

Figure 4: **Designs of Nitric Acid Cooler**

Learning Task: Few computational models of analogical reasoning have addressed learning of high-level abstractions. Birnbaum and Collins (1988) discuss the need for acquisition of abstract strategies that enable transfer of expertise from one domain to another. Their work uses explanation-based learning (EBL) techniques in failure-driven learning of abstract strategies for game playing (e.g., chess). GTMs in our work are similar to their abstract strategies in that GTMs also act as abstract plans for solving design-adaptation problems. However. Birnbaum and Collins view the abstract strategies to be useful only in accessing a relevant experience. that is. they view cases to be indexed by these abstract concepts. In contrast. in our theory. abstract models are useful in both the access and transfer stages of analogical reasoning. Moreover. in our approach learning is not only failure-driven but it also occurs from successful experiences.

Learning Method: Our model-based approach to learning is similar to Winston's model (1982) which shows that learning can be done by analogically transferring causal links in the explanation of an example to the target "concept." Our approach is also similar to explanation-based methods such as EBG (Mitchell, Keller, & Kedar-Cabelli, 1986) and EBL (DeJong & Mooney, 1986) in using explanations (SBF models) to constrain the learning of concepts. However, most of these systems assume some knowledge of the target concept *a priori*; our model-based approach attempts to "discover" them.

Also, our model-based approach differs from EBG and EBL in the kind of explanations it uses. First, while the explanations in EBG and EBL are purely causal, the explanations in SBF models are functional in nature, i.e., they not only provide a causal account but also show how causal processes result in the achievement of specific functions. Further, SBF models provide functional, structural, and behavioral decomposition of device knowledge. Second, the explanations in EBG and EBL specify how an example is an instance of a target concept while SBF models are explanations of the functioning of devices.

Conclusions

We have presented a computational model of how generic mechanisms can be learned from problem-solving experiences. We have demonstrated in the context of the design of physical devices that the generic mechanisms can be acquired incrementally from design experiences by generalization. Mental models of solutions to problems (i.e., how a given solution is *a* solution to the given problem) provide the content for learning the models of generic mechanisms. The internal organization of mental models (e.g., functional, structural, and behavioral decomposition) together with the problem-solving context provides the constraints for learning by generalization. Further, similarities between regularities in experiences determine how abstract a learned generic mechanism can be.

Elsewhere we show how our computational model also accounts for the acquisition of other types of "abstract concepts," such as mental models of physical principles, physical processes, and device prototypes (Bhatta & Goel, 1992).

Finally, from the computational model we can predict that if they have the models of specific devices, human designers can easily learn the models of generic mechanisms from their design experiences and use the learned mechanisms for making cross-domain analogies.

Acknowledgments. We thank Eleni Stroulia for her contributions to KRITIK2 and Mimi Recker for her comments on an earlier draft of this paper.

References

Bhatta, S. & Goel, A. 1992. Discovery of Physical Principles from Design Experiences. In J.M. Zytkow (Ed.), *Proceedings of the ML-92 Workshop on Machine Discovery*, 77-81. Aberdeen Scotland, UK.

Birnbaum, L. & Collins, G. 1988. The Transfer of Experience Across Planning Domains Through the Acquisition of Abstract Strategies. In J. Kolodner (Ed.), *Proceedings of the DARPA Workshop on Case-Based Reasoning*, 61-79. Clearwater Beach, FL.

Bylander, T. 1991. A Theory of Consolidation for Reasoning about Devices. *International Journal of Man-Machine Studies* 35(4):467-489.

Catrambone, R. & Holyoak, K. 1989. Overcoming Contextual Limitations on Problem-Solving Transfer. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 15(6):1147-1156.

Chandrasekaran, B., Goel, A., & Iwasaki, Y. 1993. Functional Representation As Design Rationale. *IEEE Computer*, January:48-56.

DeJong, G. & Mooney, R. 1986. Explanation-Based Learning: An Alternative View. *Machine Learning* 1(2):145-176.

Gick, M.L. & Holyoak, K.J. 1983. Schema Induction and Analogical Transfer. *Cognitive Psychology* 15:1-38.

Goel, A. 1989. Integration of Case-Based Reasoning and Model-Based Reasoning for Adaptive Design Problem Solving. Ph.D. diss., Dept. of Computer and Information Science, The Ohio State University.

Mitchell, T.M., Keller, R.M., & Kedar-Cabelli, S.T. 1986. Explanation-Based Generalization: A Unifying View. *Machine Learning* 1(1):47-80.

Sembugamoorthy, V. & Chandrasekaran, B. 1986. Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems. In J. Kolodner & C. Riesbeck (Eds.), *Experience, Memory and Reasoning*, 47-73. Hillsdale, NJ: Lawrence Erlbaum.

Stroulia, E. & Goel, A. 1992. Generic Teleological Mechanisms and their Use in Case Adaptation. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, 319-324. Bloomington, IN.

Stroulia, E., Shankar, M., Goel, A., & Penberthy, L. 1992. A Model-Based Approach to Blame-Assignment in Design. In J.S. Gero (Ed.), *Proceedings of the Second International Conference on AI in Design*, 519-537. Pittsburgh, PA.

Winston, P.H. 1982. Learning New Principles from Precedents and Exercises. *Artificial Intelligence* 19(3):321-350.

■ First European Workshop on Case-Based Reasoning ■

EWCBR-93

1 - 5 November 1993

University of Kaiserslautern (Germany)

Second Call for Papers/First Call for Participation

Preprints and Proceedings:

Accepted extended abstracts will be distributed as preprints at the workshop. Authors of accepted extended abstracts are invited to submit a long paper (about 15 pages) based on this. Accepted long papers are published within a book which will be made available after the workshop.

Organization:

The EWCBR is organized by

- the expert system section of the German society for Computer Science (GI),
- the special interest group on case-based reasoning (AK-CBR),

in cooperation with

- the European Coordinating Committee for AI (ECCAI),
- the German Chapter of the ACM,
- the Computer Science Department of the University of Kaiserslautern, an
- the German Special Research Investigation on Artificial Intelligence and Knowledge-Based Systems (SFB 314)
- the German Research Center on Artificial Intelligence (DFKI),

Organizing Committee:

Klaus-Dieter Althoff, Frank Maurer, Stefan Wess
University of Kaiserslautern
Department of Computer Science
P.O. Box 3049
D-W-6750 Kaiserslautern, Germany
Tel.: +49 631 205 3360 (3363, 3356, 3362)
Fax: +49 631 205 3357
email: ewcbr@informatik.uni-kl.de

Registration:

The registration form will be supplemented by a more detailed one with respect to payment, travelling, and accommodation aspects. Please, return your registration form to the following address:

Christine Harms
Workshop Organization EWCBR-93
c/o GMD
Postfach 1316
D-W-5205 St. Augustin 1
Germany
Tel: +49 2241 14-2473
Fax: +49 2241 14-2472 /-2618
email: christine.harms@kmx.gmd.dbp.de

Registration EWCBR-93

Last name ______________________

First Name ______________________

Institution ______________________

ZIP Code, City ______________________

Country ______________________

e-mail ______________________

Fax ______________________

Frau
Christine Harms
c/o GMD - EWCBR '93
Postfach 13 16

D-5205 Sankt Augustin 1

eneral Information:

ase-Based Reasoning is a topic which becomes more and more nportant and has raised considerable interest recently. Its various spects have been elaborated from the theoretical as well as from he practical side. It supports knowledge acquisition and problem olving, and it is related to key words like machine learning, analogy, cognitive modeling, similarity, information retrieval, among thers. Although case-based reasoning has a well defined place ithin AI-related conferences, we feel that the topic deserves a orkshop on its own, as a kick-off event for Europe.

he scientific program will include the presentation of selected apers, several invited talks, system demonstrations, as well as oster and panel sessions. An introductory course is planned for the rst day.

ubmission of Extended Abstracts/Papers:

ubmissions are invited on research covering all aspects of case-ased reasoning including (but not restricted to)

- theoretical analysis of similarity assessment
- adaptation strategies
- combination of case-based and other approaches
- cognitive modeling
- case-based learning
- relations between inductive and case-based learning/reasoning
- case-based knowledge engineering
- analogical reasoning
- relations between case-based reasoning and other approaches
- evaluation of case-based approaches
- demonstration of implemented systems
- applications of case-based reasoning

lease, contribute an extended abstract (2 to 4 pages) to one of the llowing categories:

- theoretical results
- practical/empirical results
- work in progress (posters)
- system descriptions (includes a demonstration at the workshop)

ubmissions should be made in *five copies* to the program chair:

of. Dr. M. M. Richter
niversity of Kaiserslautern
epartment of Computer Science
O.Box 3049
W-6750 Kaiserslautern, Germany
l.: +49 631 205 3360
x: +49 631 205 3357
ail: ewcbr@informatik.uni-kl.de

Program Committee:

Agnar Aamodt (Norway)
Thomas Christaller (Germany)
Klaus P. Jantke (Germany)
Janet L. Kolodner (U.S.A.)
Ramon Lopez de Mantaras (Spain)
Bruce W. Porter (U.S.A.)
Lorenza Saitta (Italy)
Gerhard Strube (Germany)

Jaime G. Carbonell (U.S.A.)
Boi V. Faltings (Switzerland)
Mark T. Keane (Ireland)
Michel Manago (France)
Bernd Neumann (Germany)
Frank Puppe (Germany)
Derek Sleeman (UK)
Walter Van de Velde (Belgium)

Invited Talks:

Mark Keane (Ireland): "Analogical Asides on Case-Based Reasoning".

Janet Kolodner (U.S.A.): "Improving Human Decision Making through Case-Based Decision Aiding".

Katharina Morik (Germany): "A Case for Inductive Learning".

Manuela Veloso (U.S.A.): "Analogical/Case-Based Reasoning in General Problem Solving".

An Introduction to Case-Based Reasoning:

The course will be held by Agnar Aamodt (University of Trondheim, Norway) and Enric Plaza (CEAB-CSIC, Spain).

Location:

EWCBR-93 will take place in the European Academy Otzenhausen.

Known at the time as the "Europe House Otzenhausen", the European Academy was founded by the European Union in May 1954 as a meeting place for promoting Franco-German reconciliation, especially among young people. With the signing of the Treaty of Rome and the birth of the European Community it was decided to develop the Europe House Otzenhausen for European Youth and adult education and as an information center. In 1969, with the subsequent increase in the volume of work, staff, and premises the Europe House Otzenhausen became the first education center to be called a European Academy. The Academy is situated in woods on the edge of the village of Otzenhausen between Trier and Kaiserslautern (south-west of Germany).

Important Dates:

- Submission deadline: 30 April 1993
- Notification of acceptance/rejection: 30 June 1993
- Camera-ready-copy: 31 July 1993
- Early Registration: 15 July 1993
- Workshop: 1-5 November 1993

I intend to submit an extended abstract for the fo lowing category:

□ theoretical/practical/empirical results

□ work in progress (posters)

□ system descriptions (demonstration)

Hardware: ______________________

Software: ______________________

____

I intend to participate at EWCBR-93. Please, send me further information. □

I intend to participate in the introductory course on case-based reasoning. □

ECCAI

german acm chapter

SFB 314

[redacted]